

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Л. В. Фетюхіна, О. А. Бутова

ТЕОРІЯ ІНФОРМАЦІЇ ТА КОДУВАННЯ

Навчально-методичний посібник

Затверджено
редакційно-видавничою
радою університету,
протокол № 1 від 20.06.2012 р.

Харків
НТУ «ХПІ»
2012

УДК 004.05
ББК 32.97я73
Ф 45

Рецензенти: *В. В. Івахно*, канд. техн. наук, проф. кафедри «Промислова та біомедична електроніка» НТУ «ХП»;
О. Г. Аврунін, канд. техн. наук, доцент кафедри «Біомедичних електронних пристроїв та систем» ХНУРЕ

Ф 45 **Фетюхіна Л. В.** Теорія інформації та кодування : навч.-метод. посібник / Л. В. Фетюхіна, О. А. Бутова – Харків: НТУ «ХП», 2012. – 68 с.

ISBN

У посібнику наведено відомості про основні поняття теорії інформації, принципи формування ефективних і перешкодостійких кодів. Надані методичні рекомендації для виконання п'ятьох лабораторних робіт на персональному комп'ютері та розрахункової роботи.

Призначено для студентів, що вивчають електроніку та інформаційні системи.

Іл. 13. Табл. 8. Бібліогр. 8 назв.

ISBN

УДК 004.05
ББК 32.97я73

© Л. В.Фетюхіна,
О. А. Бутова 2012 р.

ВСТУП

Інформація пов'язана з процесами керування в електронних системах. Такі системи здійснюють обмін інформацією, який забезпечує взаємну передачу даних між користувачами із заданою швидкістю і вірогідністю. Вони зазвичай містять: пристрої для збору інформації; пристрої для кодування/декодування інформації; пристрої для передачі інформації та зберігання інформації.

Теорія інформації та кодування вивчає вивченням кількісні характеристики інформації, передачу повідомлень по каналах зв'язку і розробку методів кодування. Студенти, що отримують знання щодо основ інформаційної технології перетворення і передачі інформації, ознайомлюються з принципами роботи, основними характеристиками і можливостями сучасних перетворювачів форми інформації, характеристиками сучасних систем передачі даних (СПД), їх різновидами та можливостями, принципами корегування інформації при помилках передачі, основами захисту інформації від несанкціонованого доступу.

У посібнику подаються теоретичні відомості, завдання та зразки виконання до п'яти лабораторних робіт. Також наводиться розрахункове завдання з прикладом його виконання. Перед виконанням лабораторної роботи студентам слід вивчити теоретичний матеріал, який містить дана робота та, за необхідності, виконати аналітичні розрахунки для виданого викладачем завдання, а на занятті отримати потрібні результати та провести їх аналіз (порівняти з теоретичними). Всі матеріали мають бути оформлені згідно з стандартами університету і подані вчасно до захисту викладачеві.

ЛАБОРАТОРНА РОБОТА 1

ПЕРЕТВОРЕННЯ ФОРМИ ІНФОРМАЦІЇ. АНАЛОГО-ЦИФРОВЕ ТА ЦИФРО-АНАЛОГОВЕ ПЕРЕТВОРЕННЯ. РОЗРАХУНОК КІЛЬКОСТІ ІНФОРМАЦІЇ

Мета роботи: вивчити принципи аналого-цифрового та цифро-аналогового перетворення. Навчитися розраховувати кількість інформації та ентропію, оцінювати похибки відтворення сигналу.

1.1. Теоретичні відомості

Первинна інформація про стан об'єкта задається датчиками об'єкта головним чином за допомогою аналогових сигналів: напруги, струму, частоти, часового інтервалу та ін. Відповідні сигнали перетворюються в цифрові еквіваленти. Якщо ж система видає цифрові коди для керування об'єктом, необхідне зворотне перетворення коду-аналогу. Такі перетворення виконують аналого-цифрові пристрої (АЦП) та цифро-аналогові пристрої (ЦАП). При цьому змінюється лише форма подання аналога-коду та коду-аналога. Перетворення форми інформації супроводжується певними похибками.

Аналого-цифрове перетворення форми інформації має дві особливості: безперервний за рівнем і у часі сигнал датчика x після перетворення являє собою окремі відліки (немає безперервності в часі); відліки подані цифровим кодом (немає безперервності за рівнем), тобто вхідний сигнал квантується і за рівнем, і за часом.

Процес заміни безперервної величини x дискретним еквівалентом називають *квантуванням сигналів за рівнем*.

Аналого-цифровий перетворювач (АЦП) має певне число пронумерованих заданих рівнів. За командою "Опитування" виконується порівняння вхідної величини $x(t)$ з наявними рівнями, а як вихідний результат видається число N – номер найбільш близького до значення $x(t)$ рівня. При цьому відстань між сусідніми рівнями може бути постійною (як у звичайній лінійці). У цьому випадку число N прямо пропорційне значенню вимірюваної величини (що кодується), а відповідні АЦП називають

лінійними чи просто АЦП. На рис. 1.1 наведений приклад квантування безперервного сигналу, результатом якого є $x^*(t)$. Відстань між сусідніми рівнями заведено називати *кроком квантування за рівнем* (ціною поділу) q , а похибка квантування e – це різниця між безперервною величиною та квантованою.

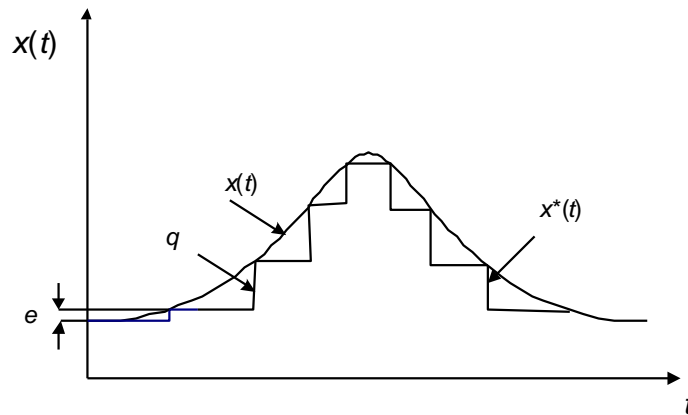


Рисунок 1.1 – Квантування сигналу

Існують АЦП, коли відстань між сусідніми рівнями q не постійна, а змінюється за визначеним законом. У цьому випадку аналого-цифрове перетворення здійснюється за описаним раніше принципом, однак зв'язок між X і N визначається нелінійною функцією $N = \varphi(X)$. Це *функціональні* або *обчислювальні* АЦП. Сигнал X , що знімається з датчика, може бути зв'язаним визначеною функціональною нелінійною залежністю з первинним параметром (висотою, тиском, швидкістю руху, силою натягу та ін.). Зазвичай при керуванні або контролі головним є визначення значення вихідного параметра, а не сигналу $x(t)$. У цьому випадку відповідний функціональний АЦП видає цифрові еквіваленти N , які прямо пропорційні значенню параметра. Така задача вирішується за допомогою звичайних АЦП, однак комп'ютер (процесорна система) веде відповідне перерахування для кожного N , отриманого на виході АЦП. На практиці головним чином застосовуються лінійні АЦП.

У системі автоматичного керування за допомогою АЦП проводиться перетворення окремих значень вихідних сигналів датчика X у цифрові еквіваленти. Це робиться в певні моменти часу, що визначаються потребами, зв'язаними з забезпеченням тієї чи іншої якості роботи таких сис-

тем. Результати перетворень (окремі відліки) можуть безпосередньо використовуватися процесорною системою або ж запам'ятовуватися для подальшої обробки (наприклад, накопичення в процесі експерименту даних та подальша їх обробка).

Процес заміни безперервної величини її окремими відліками (незалежно від точності подання відліків) називається *квантуванням сигналів за часом* або *дискретизацією*. Відліки можуть вестися через рівні або нерівні проміжки часу.

Розглянемо випадок, коли відліки отримують через рівні проміжки часу T_0 . Ці проміжки T_0 називають *кроком квантування* (дискретизації) за часом. Дискретизація, як і квантування за рівнем, зв'язана з певними похибками, що погіршують точність роботи тієї чи іншої системи. Ці похибки необхідно враховувати так, щоб їхній вплив не виходив за рамки допустимого. На рис. 1.2 наведено приклад дискретизації сигналу $x(t)$.

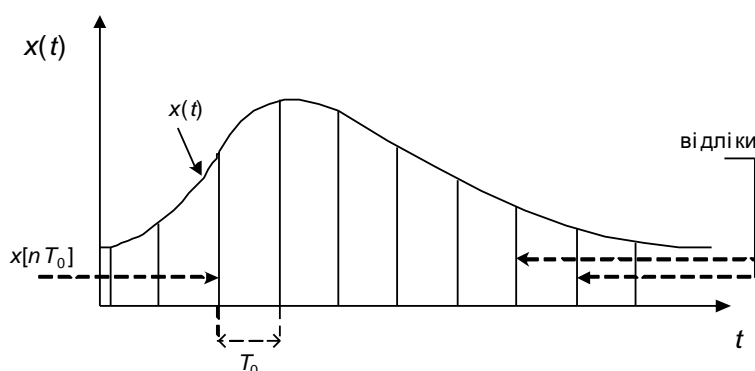


Рисунок 1.2 – Дискретизація сигналу

Вибір кроку дискретизації T_0 повинний бути таким, щоб похибка відновлення функції між відліками не перевищувала заданого значення Δx_{\max} . Це важливо і при обробці даних експерименту, що завершився, і у реальному часі, щоб не допустити неприпустимі відхилення $x(t)$ між відліками.

Для вивчення процесу дискретизації розглянемо деяку ідеальну ситуацію. Будемо вважати, що відліки беруться за допомогою АЦП, що має дуже велику швидкодію (час перетворення дорівнює нулю), саме перетво-

рення виконується без похибок, тобто відлік у момент t_k – це абсолютно точне значення $x(t_k)$. При цьому цікавими є відповіді на наступні питання:

– З якою граничною точністю можна відтворити $x(t)$ між сусідніми відліками?

– Як вибрати T_0 і при цьому гарантувати, що похибка відтворення між відліками не перевищить допустимого значення Δx_{\max} ?

На перше питання дає відповідь **теорема дискретизації** або **теорема Котельникова**, сутність якої в наступному:

Якщо безперервна функція $X(t)$ задовольняє умови Діріхле (обмежена, кусково-безперервна і має скінченне число екстремумів) і її спектр обмежений деякою частотою f_b , то вона повністю визначається послідовністю своїх значень у точках, віддалених на відстані $T_0 = 1/2f_b$ одна від одної. Аналітично теорема Котельникова виражається інтерполяційним рядом

$$x(t) = \sum_{n \rightarrow \infty} x[nT_0] \cdot \frac{\sin \omega_b(t - n \cdot T_0)}{\omega_b(t - n \cdot T_0)}. \quad (1.1)$$

Безпосередньо з цього виразу випливає, що безперервна функція з обмеженим спектром може бути подана у вигляді суми нескінченно великого числа членів, кожен з яких є добутком функції sinc від аргументу $\omega_b(t - n \cdot T_0)$ і коефіцієнта $x[nT_0]$, що визначає значення функції $x(t)$ у момент відліку.

Що стосується практичної сторони питання, то можна стверджувати, що абсолютно точного відновлення $x(t)$ не відбудеться.

По-перше, нескінченне число відліків на практиці не може бути отримано. Для визначення значення $x(t)$ між відліками використовується формула (1.1) з обмеженим числом доданків, симетрично розташованих щодо моменту часу, що розглядається. Точність буде залежати від обраного числа доданків і буде гірша на краях діапазону отриманих відліків. Рекомендоване підвищення частоти дискретизації також не може бути виправданим. Якщо сигнал знімають з виходу датчика з обмеженою частотою пропускання, то, принаймні, відома частота f_b .

По-друге, реальні процеси, що описує функція $x(t)$, скінченні у часі, а це значить, що їхні спектри нескінченні за частотою. Установка обмежувального фільтра в принципі не рятує положення, оскільки пригнічення частот вище f_b змінює саму функцію $x(t)$. Зменшення кроку дискретизації T_0 веде до зменшення похибок відновлення, але вимагає АЦП із більшою швидкодією і великим обсягом інформації, що оброблюється та запам'ятовується. Збільшення кроку дискретизації T_0 веде до зворотних наслідків. Тобто крок необхідно вибрати якомога більшим, але так, щоб похибка апроксимації не перевищувала заданого (допустимого) значення Δx_{\max} . Зрозуміло, бажано, щоб процедура відновлення була простою та не вимагала великого обсягу обчислень.

Таким чином, можна констатувати, що теорема Котельникова установлює фундаментальний факт – за відліками теоретично можна абсолютно точно відновити безперервну з обмеженим спектром функцію $x(t)$. Відновлення функції за відліками на практиці пов'язане з похибками й досить об'ємними обчислювальними роботами.

Значна частина повідомлень, особливо останнім часом, за своєю природою не є сигналами – це пакети даних, результати цифрових вимірювань різних параметрів, цифрові фотографії, текстові, графічні або інші файли тощо. Повідомлення такого типу можна подати у вигляді масивів чисел або деяких векторів. Тому інформація, що надходить в повідомленні з джерел інформації, використовує так званий первинний алфавіт, який містить m символів. Далі повідомлення переводиться в більш зручний для передачі, обробки та запису вторинний алфавіт, найчастіше двійковий, тобто *кодується*. Зрозуміло, якщо первинний алфавіт має більше двох символів, то при кодуванні двійковим алфавітом символу первинного алфавіту повинен відповідати певний набір символів вторинного (двійкового) алфавіту. Такий набір символів називають *словом* або *кодовим словом*. Якщо довжина кодового слова однакова для всіх букв первинного алфавіту, то такий код називається *рівномірним*, у іншому випадку код називають *нерівномірним*.

Кожне повідомлення характеризується ймовірністю появи, і чим нижча ймовірність, тим більше в ньому інформації. Якщо ймовірність до-

рівнює одиниці (тобто повідомлення – абсолютно достовірне), то в цьому випадку кількість інформації дорівнює нулю.

Будемо вважати що, якщо кількість типів повідомлень обмежена, наприклад, величиною m , то вони мають дискретний характер. У кожному елементарному повідомленні для його одержувача міститься деяка інформація. Визначимо кількісну міру цієї інформації і з'ясуємо, від чого вона залежить. Нехай є явище, що може мати обмежену кількість m типів проявів, а саме такі: $a_1, a_2, a_3, \dots, a_m$, до того ж відомі ймовірності $p(a_1), p(a_2), p(a_3), \dots, p(a_m)$ цих проявів. Згідно з ідеями К. Шеннона, для цього випадку підходить вираз:

$$I(a) = \log \frac{1}{p(a)} = -\log p(a). \quad (1.2)$$

Тобто кількість інформації $I(a)$, що міститься в елементарному повідомленні, є деякою функцією від ймовірності повідомлення $p(a)$. Легко пересвідчитись, що вираз (1.2) дає правильні результати і для абсолютно достовірного повідомлення, тобто при $p(a_i) = 1$ відповідно $I(a_i) = 0$. Значення $I(a_i)$ зростає при зменшенні значення $p(a_i)$.

Середня кількість інформації H для всієї сукупності повідомлень можна отримати шляхом усереднення за всіма проявами:

$$H = -\sum p_i \log p_i. \quad (1.3)$$

Основу логарифма зазвичай вибирають рівною двом. Визначена таким чином одиниця виміру інформації називається *двійковою одиницею* або *бітом інформації*. Наприклад, якщо будь-яке з елементарних повідомлень a_i може бути вибрано з алфавіту і передано з імовірністю $p(a_i) = 1/16$, то говорять, що в ньому міститься $\log_2(1/16) = 4$ біти інформації. Для визначення кількості інформації, яка міститься у повідомленні, вводять поняття *ентронії* (невизначеності) системи (об'єкта, експерименту).

Нехай тепер ансамбль повідомлень буде наданий деяким алфавітом із m типів символів, причому домовимось, що символи – рівноймовірні, самі повідомлення – статистично незалежні. Число повідомлень, які мож-

на отримати, комбінуючи m символів алфавіту по n елементів в повідомленні, дорівнює

$$I = \sum_{i=1}^m I(a_i) = \sum_{i=1}^m -\log_2 p(a_i) = \sum_{i=1}^m \log_2 n = m \log_2 n. \quad (1.4)$$

1.2. Порядок виконання роботи

Метою роботи є закріплення теоретичних знань з аналого-цифрового і цифро-аналогового перетворення сигналу та освоєння методів комп'ютерного моделювання дискретизації, квантування та кодування сигналів.

1.2.1. Вибір варіанта

Для вибору свого варіанта необхідно задати такі дані:

i_1 – дорівнює цифровому еквіваленту першої літери прізвища;

i_2 – дорівнює цифровому еквіваленту першої літери імені;

i_3 – дорівнює цифровому еквіваленту першої літери по-батькові;

j_1 – дорівнює дню народження;

j_2 – дорівнює місяцю народження;

j_3 – дорівнює порядковому номеру за списком П.І.Б. у групі.

Вихідне рівняння для безперервної функції має вигляд

$$Y(t) = \sum_{n=1}^3 y_n(t), \text{ де } y_n(t) = (i + j) \cdot f(jt). \quad (1.5)$$

Якщо j – парне, то вибирається функція $f(jt) = \cos(jt)$, в іншому випадку $f(jt) = \sin(jt)$. Таким чином, формуємо полігармонічний сигнал, який складається з трьох складових з різними амплітудами ($y_{\max} = i + j$) та частотами ($j = \omega$) в загальному випадку.

1.2.2. Завдання до лабораторної роботи

Відповідно до варіанта завдання виконати такі розрахунки:

- побудувати графіки заданих функцій;
- розрахувати частоту та період дискретизації сигналу $Y(t)$, тривалість сигналу T_c ;

– порахувати кількість відліків дискретного сигналу $Y(t)$, побудувати графік дискретного сигналу $Y(t)$ та відновлений з нього безперервний сигнал $Y(t)$;

– порахувати кількість інформації та ентропію;

– зробити оцінку похибки відновлення безперервного сигналу $Y(t)$ та побудувати графік помилки;

– зробити висновки за результатами роботи.

УВАГА! Всі розрахунки виконуються на комп'ютері.

1.3. Зразок розрахунку

Наприклад, студент Антонов Борис Володимирович народився 20.06, за списком групи третій. Тоді $i_1 = 1$; $i_2 = 2$; $i_3 = 3$; $j_1 = 20$, $j_2 = 6$, $j_3 = 3$.

Виконуємо підстановку:

$$y_1 = (1 + 20)\cos 20t = 21\cos 20t,$$

$$y_2 = (2 + 6)\cos 6t = 8\cos 6t,$$

$$y_3 = (3 + 3)\sin 3t = 6\sin 3t.$$

1) Будуємо графіки. На рис. 1.3 зображені графіки 4-х функцій, включаючи сумарну $Y(t)$. Будемо вважати, що сумарна функція $Y(t)$ являє собою безперервний сигнал, який необхідно перетворити за процедурами дискретизації та квантування.

2) Визначаємо частоту та період дискретизації сигналу $Y(t)$. Для цього необхідно визначити частоту найшвидшої функції $y_n(t)$.

Для даного прикладу це $f_b = f_1 = \frac{\omega_1}{2\pi} = \frac{20}{2 \cdot 3,14} = 3,18 \text{ Гц}$, відповідно до

неї період $T_1 = \frac{1}{3,18} = 0,31 \text{ с}$. Відповідно до теореми Котельникова обчис-

люємо частоту дискретизації з коефіцієнтом запасу $K_3 = 1,4$. $f_d = 2 \cdot 1,4 \cdot f_b = 2 \cdot 1,4 \cdot f_1 = 2 \cdot 1,4 \cdot 3,18 = 8,9 \text{ Гц}$, тоді період дискретизації

$$T_d = \Delta t = \frac{1}{f_d} = 0,11 \text{ с}.$$

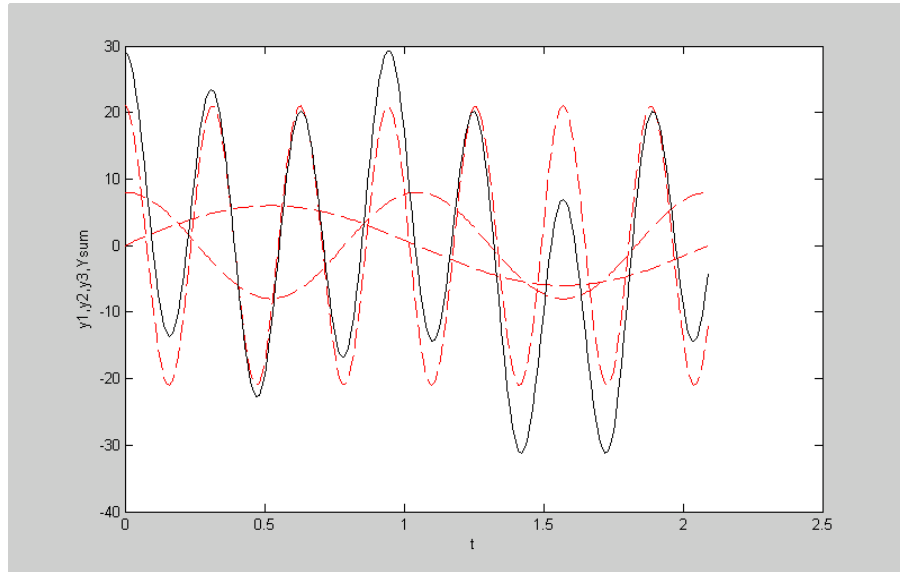


Рисунок 1.3 – Графіки 4-х функцій, у тому числі сумарної $Y(t)$

3) Визначаємо період сформованого сигналу T_c .

Будемо розглядати сигнал $Y(t)$ на періоді найповільнішої функції.

Для даного прикладу це $f_3 = \frac{\omega_3}{2\pi} = \frac{3}{2 \cdot 3,14} = 0,48 \text{ Гц}$, відповідно до неї

період $T_c = T_3 = \frac{1}{0,48} = 2,08 \text{ с}$.

4) Обираємо крок квантування дискретного сигналу $Y(t)$.

Для вибору кількості розрядів аналого-цифрового перетворення необхідно розрахувати кількість відліків дискретного сигналу:

$$Y(t) \quad N = \frac{T_c}{\Delta t}. \quad (1.5)$$

Для даного прикладу $N = \frac{2,08}{0,11} \approx 19$ інтервалів, отже, кількість розря-

дів, яка достатня для кодування, дорівнює 5 (тому що був обраний двійковий код, а $2^5 = 32$).

5) Будуємо графік дискретного сигналу $Y(t)$ (рис. 1.4). Процедuru квантування проводимо вручну (рис. 1.5). Для цього розіб'ємо весь інтервал значень дискретного сигналу $Y(t)$ рівномірно на 32 частини. Визначимо нижній ступінь квантування (мінімальне значення функції на інтервалі)

нулями (код 00000), а верхній (максимальне значення функції на інтервалі) – одиницями (код 11111).

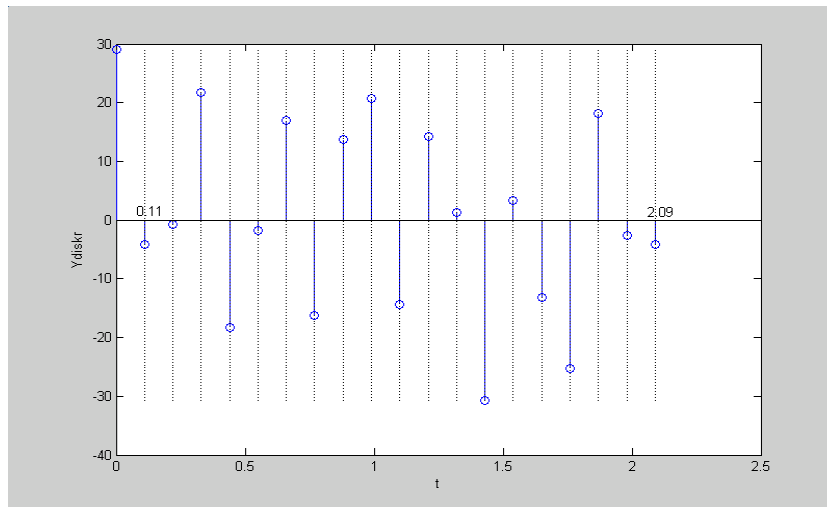


Рисунок 1.4 – Графік дискретного сигналу $Y(t)$

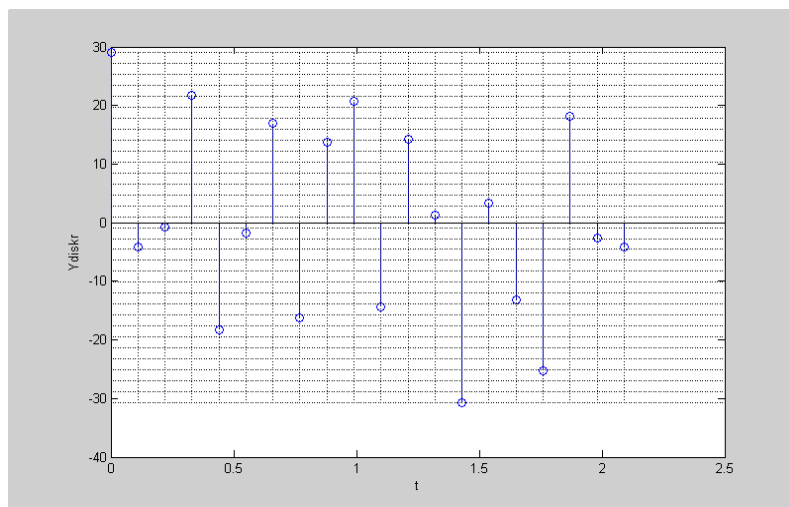


Рисунок 1.5 – Процедура квантування сигналу

Результати квантування занесемо в таблицю 1.1. Визначимо інформаційний обсяг, кількість інформації та ентропію. Для цього підрахуємо ймовірність появи кожного коду за період сигналу і заповнимо таблицю 1.2.

Таблиця 1.1 – Результати квантування

Δt	Значення y_i	Код	Δt	Значення y_i	Код
0,0	29	11111	1,1	-14,344	01001
0,11	-4,0943	01110	1,21	14,17	10111
0,22	-0,78989	10000	1,32	1,3687	10001
0,33	21,788	11011	1,43	-30,68	00000
0,44	-18,235	00111	1,54	3,3028	10010
0,55	-1,8257	10000	1,65	-13,224	01000
0,66	16,961	11000	1,76	-25, 247	00011
0,77	-16,316	01000	1,87	18,107	11001
0,88	13,817	10111	1,98	-2,6372	01111
0,99	20,766	11011	2,09	-4,1379	01110

Таблиця 1.2 – Розрахунок ймовірностей

Код	Кількість	Ймовірність	Код	Кількість	Ймовірність
00000	1	0,05	10000	2	0,1
00001	0	0	10001	1	0,05
00010	0	0	10010	1	0,05
00011	1	0,05	10011	0	0
00100	0	0	10100	0	0
00101	0	0	10101	0	0
00110	0	0	10110	0	0
00111	1	0,05	10111	2	0,1
01000	2	0,1	11000	1	0,05
01001	1	0,05	11001	1	0,05
01010	0	0	11010	0	0
01011	0	0	11011	2	0,1
01100	0	0	11100	0	0
01101	0	0	11101	0	0
01110	2	0,1	11110	0	0
01111	1	0,05	11111	1	0,05

Інформаційний обсяг джерела L визначається як

$$L = m^n = 2^5 = 32.$$

Кількість інформації I визначається за формулою

$$I = n \cdot \log_2 m = 5.$$

Ентропія джерела H

$$H = -\sum_{i=1} p_i \log_2 p_i = 1,528 \text{ біт / символ.} \quad (1.6)$$

6) Відновимо безперервний сигнал $Y(t)$ з дискретного сигналу за допомогою кусково-лінійної апроксимації. Для цього з'єднаємо відліки сигналу $Y(t)$ прямими лініями (рис. 1.6).

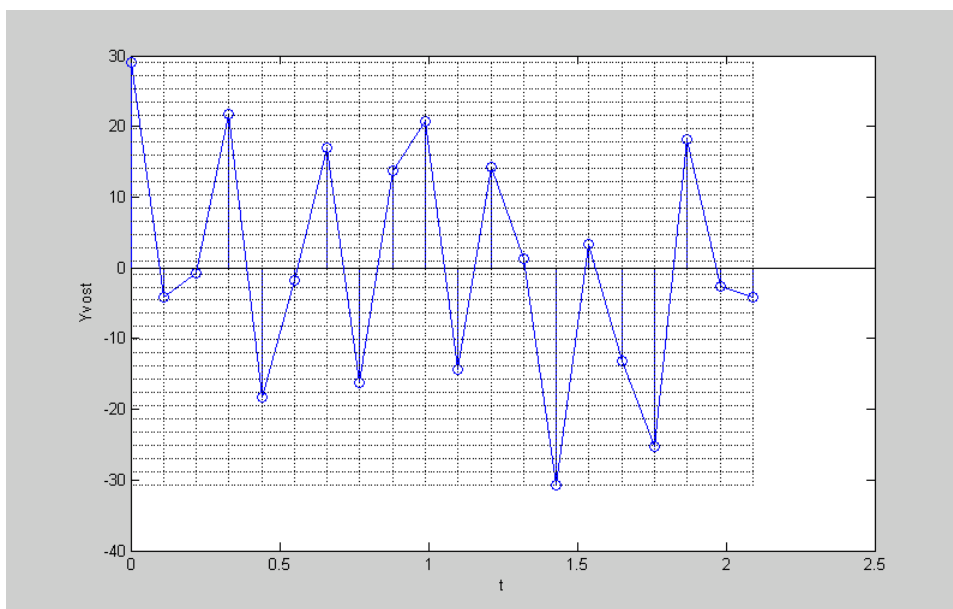


Рисунок 1.6 – Графік безперервного сигналу $Y(t)$

7) Відновимо безперервний сигнал $Y(t)$ з дискретного сигналу за допомогою теореми відліків. Для цього скористаємося такою формулою:

$$Y_{\text{відн}}(t) = \sum_{i=0}^N y_i \frac{\sin \frac{\pi}{\Delta t} (t - (\Delta t \cdot i))}{\frac{\pi}{\Delta t} (t - (\Delta t \cdot i))}, \quad (1.7)$$

де i – номер відліку;

y_i – i -те значення відліку сигналу (див. табл.1.1)

На рис. 1.7 наведено графік безперервного сигналу $Y_{\text{відн}}(t)$, який відновлений за допомогою теореми відліків.

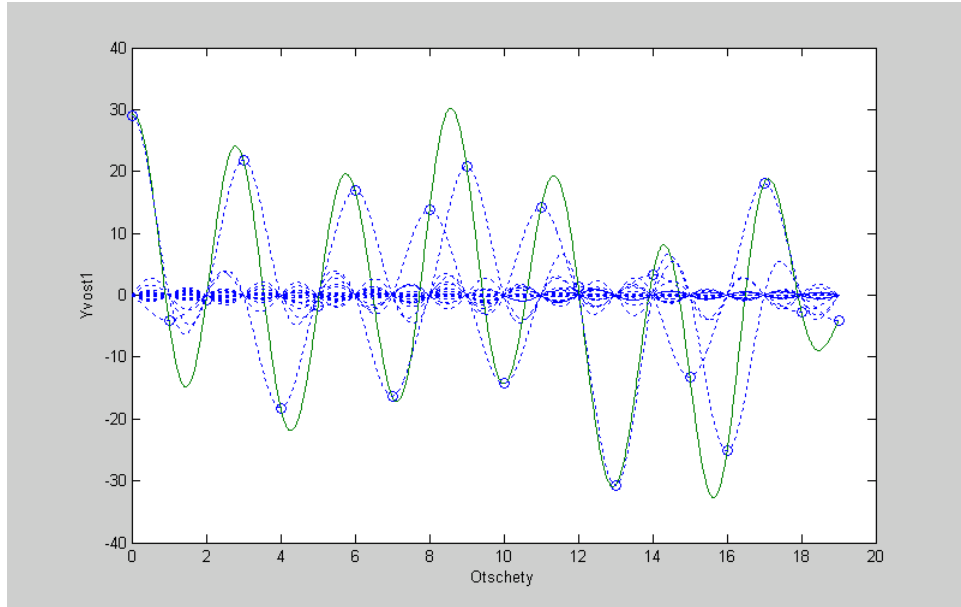


Рисунок 1.7 – Графік сигналу $Y_{\text{відн}}(t)$

8) Виконаємо кількісну оцінку похибки відновлення безперервного сигналу $Y(t)$.

Для кількісної оцінки похибки відновлення безперервного сигналу $Y(t)$ необхідно на одному рисунку зобразити первинний сигнал $Y(t)$ і сигнал, відновлений за допомогою функції відліків $Y_{\text{відн}}(t)$. Помилка відновлення $\varepsilon(t)$ розраховується як модуль різниці між цими значеннями в точках $N_1 = N_2$ і дорівнює

$$|\varepsilon(t)| = |Y(t) - Y_{\text{відн}}(t)|. \quad (1.8)$$

Середня помилка розраховується за формулою

$$\bar{\varepsilon} = \frac{1}{N_1} \sum_{i=1}^{N_1} \varepsilon_i(t). \quad (1.9)$$

Дисперсія помилки розраховується за формулою

$$\bar{D}_\varepsilon = \frac{1}{N_1} \sum_{i=1}^{N_1} (\varepsilon_i - \bar{\varepsilon})^2. \quad (1.10)$$

Для даного прикладу $\bar{\varepsilon} = 0,9595$, $\bar{D}_\varepsilon = 0,8463$.

Графіки первинного сигналу $Y(t)$, сигналу, відновленого за допомогою функції відліків, $Y_{\text{відн}}(t)$, та графік помилки відобразити разом, як на рис. 1.8.

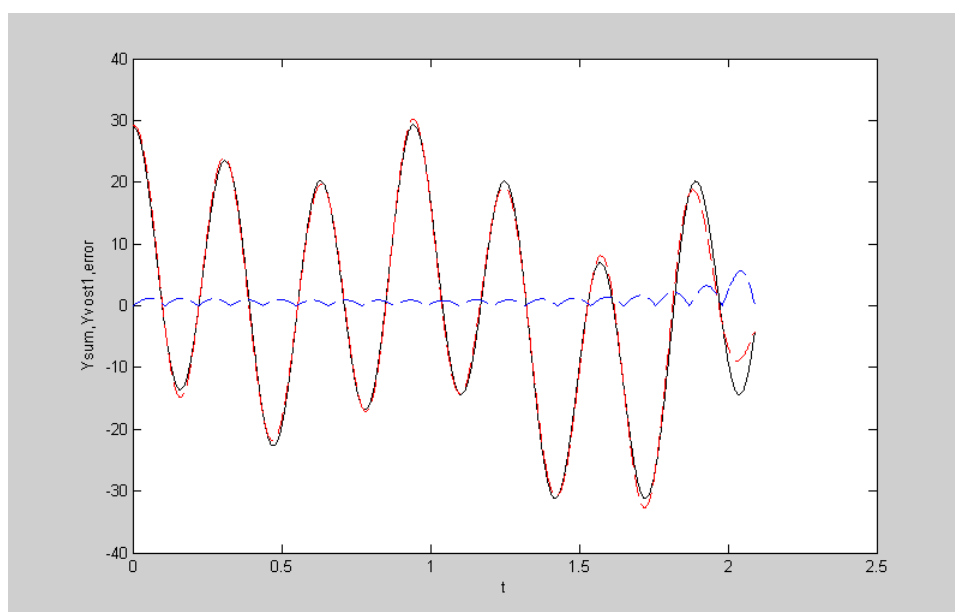


Рисунок 1.8 – Графік помилки

9) Зробимо висновки. При виконанні роботи були отримані навички з комп'ютерного моделювання процесів дискретизації, квантування та кодування сигналів. Був перевірений на практиці процес відновлення сигналів за допомогою теореми відліків.

10) Звіт повинен містити вихідні дані та складатися з розрахунків, таблиць, графіків та файлів.

ЛАБОРАТОРНА РОБОТА 2

ОПТИМАЛЬНЕ КОДУВАННЯ

Мета роботи: ознайомлення з методикою оптимального кодування Шеннона-Фано та Хаффмана.

2.1. Теоретичні відомості

Повідомлення, що передаються з використанням систем передачі інформації (мова, музика, зображення і т. ін.), у більшості своїй призначені для безпосереднього сприйняття органами почуттів людини і погано пристосовані для їх ефективної передачі по каналах зв'язку. Тому вони в процесі передачі, як правило, підлягають кодуванню.

Кодуванням інформації називається операція перетворення повідомлень в певну послідовність сигналів, а зворотна операція, що відновлює повідомлення за прийнятими сигналами, називається *декодуванням*. На практиці широко використовуються різні коди: двійковий код, двійково-десятковий код, шістнадцятковий код, код Бодо, американський стандартний код обміну інформації (ASCII) і т. ін.

Кодування повідомлень може мати на меті різні цілі. Це може бути кодування з метою засекречування переданої інформації. Іншим прикладом кодування може служити перетворення дискретних повідомлень з одних систем числення в інші (з десятикової в двійкову, вісімкову і т. п.; з позиційної в позиційну; перетворення буквеного алфавіту в цифровий і т. д.). Кодування в каналі або перешкодостійке кодування інформації може бути використано для зменшення кількості помилок, що виникають при передачі по каналу з перешкодами.

Кодування повідомлень може проводитися з метою скорочення обсягу інформації та підвищення швидкості її передачі або скорочення смуги частот, необхідних для передачі. Таке кодування називають *економним*, *беззбитковим* або *ефективним* кодуванням, а також *стискуванням даних*.

Оптимально закодованим будемо вважати такий код, при якому на передачу повідомлень витрачається мінімальний час. Якщо на передачу

кожного елементарного символу (0 або 1) витратити однаковий час, то оптимальним буде такий код, який має мінімально можливу довжину.

Принципи побудови оптимальних кодів:

1. Кожний елементарний символ повинен переносити максимальну кількість інформації, для цього необхідно, щоб елементарні символи (0 і 1) у закодованому тексті зустрічалися в середньому однаково часто, тобто були рівноймовірними. Ентропія в цьому випадку буде максимальною.

2. Необхідно буквам первинного алфавіту, що мають більшу ймовірність, присвоювати більш короткі кодові слова вторинного алфавіту. В результаті зменшується середня довжина кодової групи в порівнянні з випадком рівномірного кодування.

Припустимо, що є джерело, яке дає такі типи повідомлень: $a_1, a_2, a_3, \dots, a_q$. Ймовірності цих повідомлень задані: $p(a_1), p(a_2), p(a_3), \dots, p(a_q)$. Кожне з цих повідомлень потрібно закодувати двійковим кодом, витративши на кодування k -го повідомлення n_k розрядів (n_k можуть бути різними), до того ж двійковий розряд передається за час τ . Тоді швидкість передачі R можна визначити як

$$R = H(a) / \bar{\tau}, \quad (2.1)$$

де $H(a) = \sum_{k=1}^q p(a_k) \log_2 p(a_k)$ – ентропія джерела повідомлень;

$\bar{\tau}$ – середній час передачі кодової комбінації, який обчислюється так:

$$\bar{\tau} = \tau \sum_{k=1}^q n_k p(a_k), \quad (2.2)$$

де τ – час передачі елемента коду.

Як бачимо, чисельник формули (2.1) залежить виключно від статистичних даних якостей джерела, а знаменник формули, крім того, і від способу кодування (які значення прийняті для n_k) та якостей каналу (τ).

Очевидно, що більш ймовірні повідомлення передаються за короткий час, менш ймовірні – за довший.

При оптимальному кодуванні швидкість передачі інформації має наближатись до пропускної спроможності.

Одним з конкретних оптимальних кодів є так званий код Шеннона-Фано. Алгоритм його побудови такий:

- усі типи повідомлень записуються в стовпчик у порядку зменшення їх ймовірностей;
- стовпчик розділяється на дві частини з сумарно приблизно рівними ймовірностями та для всіх повідомлень верхньої частини перший розряд записують одиницю, а для нижньої частини – нуль;
- за тим же принципом розділяється на дві частини кожна з одержаних раніше частин і таким же методом записуються інші цифри повідомлень.

Такий процес поділу та кодування проводиться доти, доки в обох частинах не стане по одному елементу. Ясно, що код буде нерівномірним, але він буде оптимальним.

У нерівномірних кодах при декодуванні виникають труднощі з виявленням межі повідомлень. Для виключення помилок, як правило, використовують спеціальні роздільні знаки, як наприклад, у коді Морзе, а це призводить до зменшення швидкості передачі й деякого ускладнення пристроїв.

Код Шеннона відрізняється так званою властивістю *префіксності*: жодна кодова комбінація символу не входить як початкова ділянка у код іншого символу. Якщо при побудові коду умова префіксності не виконується, то коди окремих символів неможливо розділити в загальному потоці.

Розглянута методика Шеннона-Фано не завжди приводить до однозначної побудови коду, оскільки при розбитті на підгрупи можна зробити більшою за ймовірністю як верхню, так і нижню підгрупи.

Методика Хаффмена гарантує однозначну побудову коду з найменшим для даного розподілу ймовірності середнім числом символів на букву. Для двійкового коду методика зводиться до наступного.

На першому етапі символи упорядковують за зменшенням ймовірності, а потім виконують декілька кроків «об'єднання», на кожному з яких підсумовуються ймовірності, що є найменшими (дві останні букви об'єднують в одну допоміжну букву, якій приписують сумарну ймовірність). Далі стовпчик ймовірності сортирується заново за зменшенням.

На другому етапі будується «дерево коду», гілки якого відображають у зворотному порядку процес «об'єднання ймовірностей». При побудові

дерева приймається правило відповідності більшій ймовірності одному з напрямів гілки (наприклад, «лівому») і значенню біта коду (наприклад, «1»). Ланцюжки бітів від «кореня» до кінця кожної гілки відповідають кодам символів.

Звернемо увагу на той факт, що як для коду Хаффмена, так і для коду Шеннона-Фано середня кількість двійкових символів, що припадає на символ джерела, наближається до ентропії джерела, але не дорівнює їй. Даний результат є наслідком теореми кодування без шуму для джерела (першої теореми Шеннона), яка стверджує:

Будь-яке джерело можна закодувати двійковою послідовністю при середній кількості двійкових символів на символ джерела, як завгодно близькому до ентропії, і неможливо домогтися середньої довжини коду l_{cp} , меншої, ніж ентропія H_1 . При розрахунках в лабораторній роботі слід використовувати такі формули.

Середнє число двійкових символів на літеру (довжина коду):

$$l_{cp} = -\sum p_i \cdot l_i, \quad (2.3)$$

l_i – кількість розрядів в i -му коді.

Ентропія первинного коду:

$$H_1 = -\sum p_i \cdot \log_2 p_i. \quad (2.4)$$

Ймовірності появи одиниць і нулів у коді:

$$p(1) = \frac{\sum p_i \cdot z_{i1}}{l_{cp}}; \quad p(0) = 1 - p(1), \quad (2.5)$$

z_{i1} – кількість одиниць в i -му коді.

Ентропія вторинного коду

$$H_2 = -(p(0) \cdot \log_2 p(0) + p(1) \cdot \log_2 p(1)). \quad (2.6)$$

Ефективність оптимальних кодів оцінюють за допомогою коефіцієнта статистичного стиснення, який характеризує зменшення кількості двій-

кових знаків на символ повідомлення при застосуванні оптимального коду в порівнянні з застосуванням методів нестатистичного кодування

$$K_{CC1} = \frac{H_{\max 1}}{l_{cp}} \quad (2.7)$$

і коефіцієнта відносної ефективності первинного коду

$$K_{BE1} = \frac{H_1}{l_{cp}}, \quad (2.8)$$

який показує, наскільки використовується статистична надмірність переданого повідомлення.

Коефіцієнт статистичного стиснення і відносної ефективності вторинного коду визначаються аналогічно:

$$K_{CC2} = \frac{H_{\max 2}}{l_{cp}}; K_{BE2} = \frac{H_2}{l_{cp}}. \quad (2.9)$$

Для визначення кількості «зайвої» інформації, яка закладена в структурі алфавіту або в природі коду, вводиться поняття *надмірності*. Надмірність, з якою ми маємо справу в теорії інформації, не залежить від змісту повідомлення і заздалегідь відома зі статистичних даних. Інформаційна надмірність показує відносне недовантаження на символ алфавіту і є безрозмірною величиною:

$$D = 1 - \mu = 1 - \frac{H_2}{H_{\max 2}}, \quad (2.10)$$

де μ – коефіцієнт стискання (відносна ентропія).

Розглянемо процедуру ефективного кодування повідомлень, утворених за допомогою алфавіту, що складається всього із двох знаків z_1 і z_2 , ймовірностями появи яких відповідно $p(z_1) = 0,9$ і $p(z_2) = 0,1$. Оскільки ймовірності не рівні, то послідовність з таких букв матиме надмірність. Однак при кодуванні ніякого ефекту не отримаємо. Дійсно, на передачу кожної літери потрібен символ або 1, або 0, у той час як ентропія повідомлення H_1 , що розрахована за формулою Шеннона, дорівнює 0,47. При кодуванні блоків, що містять по дві літери, отримаємо середнє число символів на блок — 1,29 і на букву — 0,645 (оскільки літери статистично не зв'язані, ймовірності блоків визначаються як добуток ймовірностей складових

літер). Кодування блоків, що містять по три літери, дає ще більший ефект. Але теоретичний мінімум $l_{cp} = H_1 = 0,47$ може бути досягнутий при кодуванні блоків, що включають нескінченне число літер.

2.2. Порядок виконання роботи

2.2.1. Вибір варіанта

j_1 = день народження – відповідає середньому часу виробітку символу на виході джерела в мс;

j_2 = порядковий номер за списком – відповідає варіантам завдання. Вхідними є символи первинного алфавіту x_i та їхні ймовірності $p(x_i)$.

2.2.2. Послідовність дій

– Необхідно зробити кодування методом Шеннона-Фано і Хаффмена та розрахувати коефіцієнти ефективності коду, оцінити швидкість передачі та пропускну спроможність.

– Зробити порівняльний аналіз двох методів кодування. Отримані результати занести в таблицю.

– Виконати процедуру блокового кодування методом Шеннона-Фано та Хаффмена для двох і трьох символних блоків.

– Порівняти результати.

– Зробити висновки.

УВАГА! Всі розрахунки виконуються на комп'ютері.

2.3. Зразок виконання завдання

1) Згідно з завданням маємо ймовірності десяти символів.

Таблиця 2.1 –Таблиця ймовірностей символів

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
0,26	0,14	0,13	0,13	0,11	0,09	0,07	0,05	0,01	0,01

Використовуючи алгоритм поділу, виконуємо кодування за методом Шеннона-Фано. Результати разом з розрахунками ентропії заносимо в таблицю 2.2.

Таблиця 2.2 – Кодування за методом Шеннона-Фано

x_i	p_i	$p_i \cdot \log_2 p_i$	Код	H_1
x_1	0,26	0,505288	11	
x_2	0,14	0,39711	101	
x_3	0,13	0,382644	100	
x_4	0,13	0,382644	011	
x_5	0,11	0,350287	010	
x_6	0,09	0,312654	001	
x_7	0,07	0,268555	0001	
x_8	0,05	0,216096	00001	
x_9	0,01	0,066439	000001	
x_{10}	0,01	0,066439	000000	

Кодування за методом Хаффмена реалізуємо за допомогою алгоритму стиснення. Результати заносимо до таблиці 2.3.

Таблиця 2.3 – Кодування за методом Хаффмена

x_i	p_i										Код
x_1	0,26	0,26	0,26	0,26	0,26	0,26	0,28	0,46	0,54	1	10
x_2	0,14	0,14	0,14	0,14	0,2	0,26	0,26	0,28	0,46		111
x_3	0,13	0,13	0,13	0,14	0,14	0,2	0,26	0,26			011
x_4	0,13	0,13	0,13	0,13	0,14	0,14	0,2				010
x_5	0,11	0,11	0,11	0,13	0,13	0,14					001
x_6	0,09	0,09	0,09	0,11	0,13						000
x_7	0,07	0,07	0,07	0,09							1101
x_8	0,05	0,05	0,07								11001
x_9	0,01	0,02									110001
x_{10}	0,01										110000

Для наочності побудуємо кодове дерево. Переміщаючись по кодовому дереву зверху вниз, можна записати для кожної букви відповідну їй кодову комбінацію.

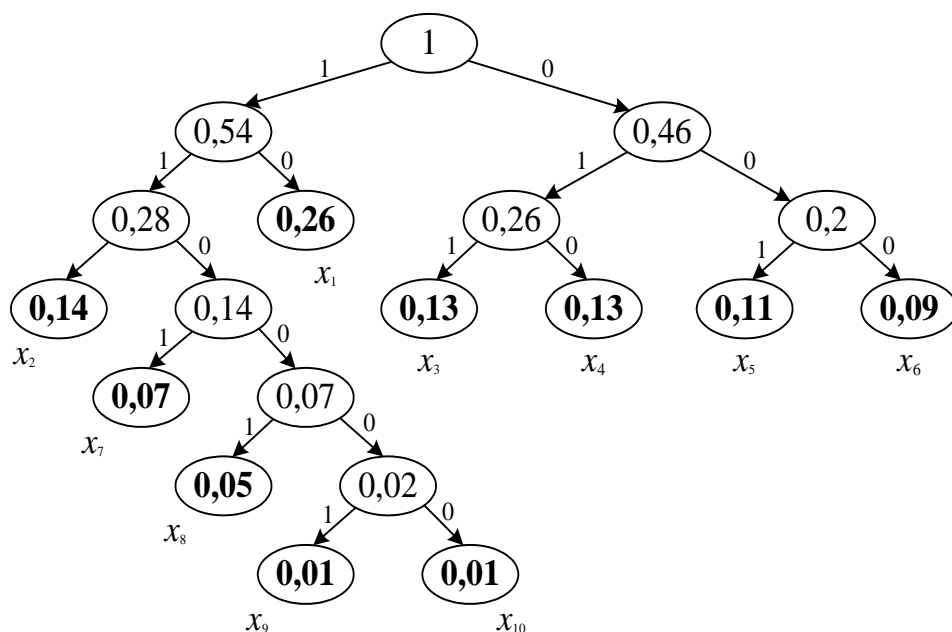


Рисунок 2.1 – Кодове дерево

Проводимо порівняльний аналіз двох методик. Розрахунок проведемо за формулами (2.3-2.10).

Результати наведемо в таблиці 2.7.

Таблиця 2.7 – Порівняльний аналіз двох методів

Параметр	Метод кодування		Граничне значення
	Шеннона-Фано	Хаффмена	
1	2	3	4
I_{cp}	2,97	2,97	2,948
H_1	2,948	2,948	3,32
H_2	0,999	0,999	1
K_{BE1}	0,9925	0,9925	1
K_{CC1}	1,01	1,01	—
K_{BE2}	0,337	0,335	—
K_{CC2}	0,3367	0,3367	—
D	0,0004	0,0036	0
$p(0)$	0,488	0,465	0,5

2) Здійснимо блокове кодування. Згідно з варіантом вхідними даними є символи первинного алфавіту і їхні ймовірності: $p(a_1) = 0,83$;

$p(a_2) = 0,17$. Виконаємо кодування для блоків, що мають один, два та три символи у повідомленні. Результати зведемо до таблиць 2.4-2.6.

Таблиця 2.4 – Розрахунок односимвольних блоків

Блок	p_i	$p_i \cdot \log_2 p_i$	Код	l_{cp}	H_1	$p(0)$	$p(1)$	H_2	K_{CC}	K_{BE}	D
a_1	0,83	0,223118	1								
a_2	0,17	0,434587	0								

Таблиця 2.5 – Розрахунок двосимвольних блоків

Блок	p_i	$p_i \cdot \log_2 p_i$	Код	l_{cp}	H_1	$p(0)$	$p(1)$	H_2	K_{CC}	K_{BE}	D
a_1a_1	0,6889	0,370376	1								
a_1a_2	0,1411	0,398637	0 1								
a_2a_1	0,1411	0,398637	0 0 1								
a_2a_2	0,0289	0,14776	0 0 0								

Таблиця 2.6 – Розрахунок трихсимвольних блоків

Блок	p_i	$p_i \cdot \log_2 p_i$	Код	l_{cp}	H_1	$p(0)$	$p(1)$	H_2	K_{CC}	K_{BE}	D
$a_1a_1a_1$	0,571787	0,461118	1								
$a_1a_1a_2$	0,117113	0,362351	0 1 1								
$a_1a_2a_1$	0,117113	0,362351	0 1 0								
$a_2a_1a_1$	0,117113	0,362351	0 0 1								
$a_2a_1a_2$	0,023987	0,129089	0 0 0 1 1								
$a_2a_1a_2$	0,023987	0,129089	0 0 0 1 0								
$a_2a_2a_1$	0,023987	0,129089	0 0 0 0 1								
$a_2a_2a_2$	0,004913	0,037679	0 0 0 0 0								

Повторюємо аналогічні розрахунки для методики Хаффмена.

$p(1)$	0,512	0,535	0,5
$R(\tau=1\text{мс})$	992	992	1114

Висновки. Коди, що були отримані, дуже близькі до оптимальних. Розглянуті методики кодування дають практично однакові результати.

ЛАБОРАТОРНА РОБОТА 3

ПЕРЕШКОДОСТІЙКЕ КОДУВАННЯ. ЛІНІЙНІ ГРУПОВІ КОДИ

Мета роботи: дослідити побудову та можливості корегування лінійних систематичних групових кодів.

3.1. Теоретичні відомості

Згідно з теоремою Шеннона існує теоретична можливість вести передачу даних по каналу з перешкодами з як завгодно малою ймовірністю помилки при процедурах кодування та декодування. Надійність передачі повідомлень підвищується за рахунок введення штучної надмірності.

Побудова перешкодостійких кодів в основному пов'язана з додаванням до вихідної комбінації контрольних m -символів (рис. 3.1). Закодована комбінація складається з n -символів. Ці коди часто називають (n,k) - коди.

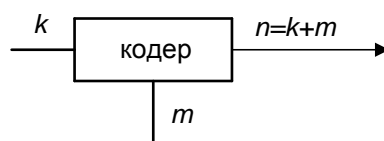


Рисунок 3.1 – Отримання (n,k) - кодів

При передачі повідомлення є певна ймовірність того, що деяка буква вторинного алфавіту буде передана неправильно (одиниця замість нуля або навпаки). Для виправлення таких помилок використовують код з виявленням помилок або код з виправленням помилки. В першому випадку кожне прийняте кодове слово перевіряється на наявність помилки. У випадку наявності помилки повідомляється передавачу, і він повторно передає це слово. У другому випадку кодове слово виправляють на стороні приймача. Слід зауважити, що ці коди можуть виявляти або виправляти лише певну кількість помилок у кодовому слові.

Найпростіший спосіб виявлення однієї помилки в кодовому слові – це доповнення до слова ще одного біта, який називають *бітом перевірки на парність*. Кількість одиниць у кодовому слові (з урахуванням додаткового біта) має бути парною. Це можна зробити, маніпулюючи додатковим бі-

том. Якщо після приймання кількості одиниць виявиться непарною, то це свідчить про помилку.

Зрозуміло, що при парній кількості помилок цей метод не виявить помилки. При кодуванні з виявленням і виправленням помилок використовують таку величину, як *відстань між кодами*. Це кількість розрядів, в яких кодові слова мають різні букви. Приміром, кодова відстань між кодами 001101 і 001000 дорівнює двом, бо вони відрізняються лише в першому і третьому розрядах, рахуючи справа. Звичайно відстань між кодами визначають, використовуючи логічну операцію *XOR* (додавання за модулем два). Ця операція позначається знаком \oplus і дає результат 1, якщо один з операндів дорівнює нулю, а другий – одиниці, в інших випадках результат дорівнює нулю. Додаємо за модулем 2 два кодових слова та рахуємо кількість одиниць у результаті. Отримуємо відстань між цими кодами. Можна для наочності використати геометричне подання коду (рис. 3.2).

Візьмемо куб, довжина ребер в якому дорівнює одиниці. Кодові слова – це вершини куба, а відстань між ними – це мінімальна кількість ребер, яку треба перейти, щоб дістатися від одного кодового слова до другого. Приміром, відстань між словами 001 і 010 дорівнює двом (проходимо два ребра). Або $001 \oplus 010 = 011$ – дві одиниці в результаті зазначають, що відстань між кодами дорівнює двом.

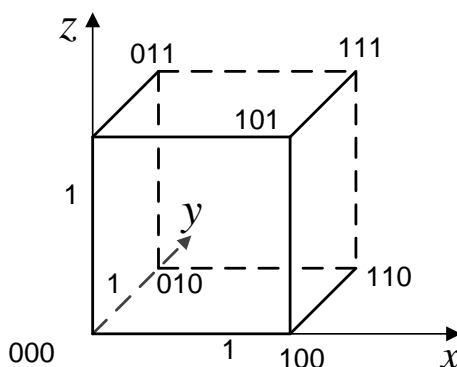


Рисунок 3.2 – Подання двійкових кодів за допомогою куба

Мінімальна відстань між вершинами визначається мінімальною кількістю ребер, що знаходяться між вершинами. Ця відстань називається *ковою* (або *хеммінговою*) і позначається буквою *d*. Інакше, *кодова відстань* – це те мінімальне число елементів, в яких одна кодова комбінація відрізняється від іншої.

няється від іншої. Як зазначалося вище, для визначення кодової відстані досить порівняти дві кодові комбінації за модулем 2.

Так, склавши дві комбінації

$$\begin{array}{r} 10110101101 \\ \oplus 11001010101 \\ \hline 01111111000 \end{array},$$

визначимо, що відстань між ними $d = 7$. Для коду з $N = 3$ вісім кодових комбінацій розміщуються на вершинах тривимірного куба. Такий код має кодову відстань $d = 1$, і для передачі використовуються всі вісім кодових комбінацій 000, 001..., 111. Такий код є не перешкодостійким, він не в змозі виявити помилку. Якщо виберемо комбінації з кодовою відстанню $d = 2$, наприклад, 000, 110, 101, 011, то такий код дозволить виявляти однократні помилки. Назвемо ці комбінації дозволеними, призначеними для передачі інформації. Всі інші 001, 010, 100, 111 – заборонені. Будь-яка одинична помилка призводить до того, що дозволена комбінація переходить в найближчу, заборонену комбінацію. Отримавши заборонену комбінацію, ми виявимо помилку. Виберемо далі вершини з кодовою відстанню $d = 3$. Такий код може виправити одну одиничну помилку або виявити дві помилки. Таким чином, збільшуючи кодову відстань, можна збільшити перешкодостійкість коду.

Узагальнивши наведене вище, можна сказати, що для виявлення r помилок відстань між дозволеними кодами повинна бути $d = r + 1$, а для виправлення s помилок відстань повинна бути $d = 2 \cdot s + 1$. Зрозуміло, що $r > s$.

У загальному випадку кодова відстань визначається за формулою

$$d = s + r + 1. \quad (3.1)$$

Більшість кодів, що виправляють помилки, є лінійними кодами.

Лінійні коди – це такі коди, в яких контрольні символи утворюються шляхом лінійної комбінації інформаційних символів. Крім того, коди, що виправляють помилки, є також груповими кодами.

Групові коди G_n – це такі коди, які мають одну основну операцію. При цьому потрібно дотримуватися умови замкнутості (тобто, при складанні двох елементів групи виходить елемент, що належить цій же групі).

Число розрядів у групі не повинно збільшуватися. Цю умову задовольняє логічна операція *XOR*. У групі, крім того, має бути нульовий елемент.

Для побудови коду, який здатний виявляти і виправляти одиничну помилку, необхідне число контрольних розрядів становитиме

$$n - k \geq \log_2(n + 1). \quad (3.2)$$

Завдання побудови перешкодостійких кодів полягає в тому, щоб забезпечити відстань між словами не менше d . Для цього матриця $G(n, k)$, що породжує код, будується за таким правилом:

Число строк дорівнює k , а число стовпчиків дорівнює $n = k + m$.

$$G(n, k) = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1k} & p_{11} & p_{12} & \dots & p_{1m} \\ a_{21} & a_{22} & \dots & a_{2k} & p_{21} & p_{22} & \dots & p_{2m} \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot & \dots & \cdot \\ a_{k1} & a_{k2} & \dots & a_{kk} & p_{k1} & p_{k2} & \dots & p_{km} \end{vmatrix}. \quad (3.3)$$

Матриця $G(n, k)$, що породжує код, складається з інформаційної (I_{kk}) і перевіркової (R_{km}) матриць. Число стовпців I -матриці дорівнює m , число стовпців R -матриці дорівнює k . Вона є стислим описом лінійного коду і може бути подана в канонічній формі:

$$G(n, k) = |I_{kk} R_{km}| = |R_{km} I_{kk}|. \quad (3.4)$$

За інформаційну I -матрицю зручно використовувати одиничну матрицю, ранг якої визначається кількістю інформаційних розрядів. Рядками одиничної матриці є незалежні комбінації, тобто їх попарне підсумовування за модулем два не приводить до нульового рядка. Стовпці додаткової R -матриці визначають правила формування перевірок. Число одиниць в кожному рядку додаткової матриці повинне задовольняти умову: $r_l \geq d - W_l$ ($W_l = 1$, оскільки одинична матриця). Слід пам'ятати, що визначене число одиниць є числом суматорів за модулем два в шифраторі і дешифраторі, і чим їх більше, тим складніша апаратура.

Рядки перетворювальної матриці являють собою k комбінацій коду. Таким чином, інформаційна частина залишається без змін, а коригувальні

розряди визначаються шляхом підсумовування за модулем два тих рядків перевірної матриці, номери яких збігаються з номерами розрядів, що містять одиницю в інформаційній частині коду.

Декодування та корекція помилок у лінійних кодах пов'язані з виконанням перевірок. Для кожної конкретної матриці існує своя єдина система перевірок. Перевірки проводяться за таким правилом: у першу перевірку разом з перевірним символом (бітом) входять інформаційні розряди, відповідні до одиниць першого стовпця перевірної матриці (R), у другу – другий перевірний символ і інформаційні символи, відповідні до одиниць другого стовпця і т.д. Число перевірок дорівнює числу перевірних символів m . Рядки перевірної матриці визначають правила формування перевірок. У результаті перевірок утворюється перевірочний вектор $S_1, S_2, S_3 \dots$ – *синдром*. Якщо число одиниць у розрядах, що перевіряються, парне, то значення S дорівнює 0. Якщо вага синдрому дорівнює нулю, то прийнята комбінація безпомилкова. В іншому випадку отриманий синдром порівнюємо зі стовпцями матриці і визначаємо розряд, у якому відбулася помилка, номер стовпця дорівнює номеру помилкового розряду. Для виправлення помилки хибний біт необхідно інвертувати.

Вид синдрому може бути визначений за допомогою перевірної матриці $H(n, k)$:

$$H(n, k) = \left| R_{mk}^T I_{mm} \right|, \quad (3.5)$$

де R_{mk}^T – транспонована перевірна матриця (поміняти рядки на стовпці);

I_{mm} – одинична матриця, число стовпців якої дорівнює числу перевірних розрядів.

Іноді важливо мати можливість автоматично виправляти помилки, що виникли в будь-якому кодовому слові. Це, наприклад, необхідно при зберіганні даних у напівпровідниковій пам'яті, де помилки з'являються в різних розрядах незалежно, а це значить, що ймовірність однократної помилки на кілька порядків вища, ніж у дво-, трикратної і т.д. Таким чином, головне завдання – забезпечити виправлення саме однократних помилок. Це завдання вирішує код Хеммінга.

Код Хеммінга належить до класу лінійних кодів і являє собою *систематичний* код – код, у якому інформаційні та контрольні біти розташовані на строго певних місцях у кодовій комбінації. Існують різні методи реалізації коду Хеммінга та кодів, які є модифікацією коду Хеммінга.

Розглянемо алгоритм побудови коду для виправлення одиначної помилки.

1. За заданою кількістю інформаційних символів k або інформаційних комбінацій $N = 2^k$, використовуючи співвідношення

$$n = k + m, \quad 2^n \geq (n + 1)2^k, \quad 2^m \geq n + 1, \quad (3.6)$$

де $m = \lceil \log_2 \{(k + 1) + \lceil \log_2(k + 1) \rceil\} \rceil$, (3.7)

обчислюємо основні параметри коду n і m .

2. Визначаємо робочі та контрольні позиції кодової комбінації. Нумери контрольних позицій визначаються за законом 2^i , де $i = 1, 2, 3, \dots$ тобто вони дорівнюють 1, 2, 4, 8, 16, ..., рахуючи зліва, а інші позиції є робочими.

3. Визначаємо значення контрольних розрядів (0 або 1) за допомогою багаторазових перевірок кодової комбінації на парність. Кількість перевірок дорівнює кількості перевірних символів $m = n - k$. У кожну перевірку включається один контрольний і певні перевірні біти. Якщо результат перевірки дає парне число, то контрольному біту присвоюється значення – 0, а якщо ні, то – 1. Нумери інформаційних бітів, що включаються в кожну перевірку, визначаються за двійковим кодом натуральних n -чисел розрядністю m або за допомогою перевірної матриці H_m , стовпці якої подають запис у двійковій системі всіх цілих чисел від 1 до $2^k - 1$, перерахованих у порядку зростання. Для $m = 3$ перевірна матриця має вигляд:

$$H = \begin{vmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{vmatrix}.$$

Кількість розрядів m визначає кількість перевірок.

У першу перевірку включають коефіцієнти, що містять одиницю в першому (нижній рядок) розряді. Тобто, якщо визначити, що b_i – це i -тий

біт рядка, це будуть b_1, b_3, b_5 і т.д. У другу перевірку включають коефіцієнти, що містять 1 в другому розряді, тобто b_2, b_3, b_6 і т.д. У третю перевірку – коефіцієнти, які містять 1 у третьому розряді і т.д.

Для виявлення і виправлення помилки складаються аналогічні перевірки на парність контрольних сум, результатом яких є двійкове $(n-k)$ -розрядне число – *синдром* і вказує на положення помилки, тобто номер помилкової позиції, який визначається за двійковим записом числа або за перевіркою матрицею. Для опису помилок, що виникають у каналі, використовують вектор помилки e – двійкова послідовність довжиною n з одиницями в тих позиціях, в яких відбулися помилки. Так, вектор помилки $e = (0\ 0\ 0\ 1\ 0\ 0\ 0)$ означає одноразову помилку в четвертій позиції (четвертому біті). Порядок рахування розрядів зліва направо.

3.2. Порядок виконання роботи

3.2.1. Вибір варіанта

j_1 = день та місяць народження (сума цифр) – відповідає позиції помилкового символу та дорівнює $j_1 \bmod (n)$, де n – повна кількість символів, що передається – для завдання 1.

j_2 = рік народження (сума цифр) – відповідає позиції помилкового символу та дорівнює $j_2 \bmod (n)$ – для завдання 2;

j_3 = порядковий номер за списком – відповідає варіантам завдання.

3.2.2. Послідовність дій

Відповідно до варіанта завдання виконати такі дії:

- показати процес кодування та декодування слова;
- процес виправлення одиничної помилки для інформаційного слова, що передається;
- зробити висновки.

3.3. Зразок розрахунку завдання

3.3.1. Зразок виконання завдання 1

Матриця, що утворює код, має вигляд:

$$G(7, 4) = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{vmatrix}$$

Тобто, маємо код (7, 4). Визначимо комбінації коду, що корегує помилки. Для заданого числа інформаційних розрядів $k = 4$ число кодів комбінацій дорівнює $N = 2^k = 2^4 = 16$. Усі можливі комбінації 4 – значного коду (старший біт справа, відповідно до їх надходження на вхід декодера):

- | | | | |
|---------|---------|----------|-----------|
| 1) 0000 | 5) 0010 | 9) 0001 | 13) 0011 |
| 2) 1000 | 6) 1010 | 10) 1001 | 14) 1011 |
| 3) 0100 | 7) 0110 | 11) 0101 | 15) 0111 |
| 4) 1100 | 8) 1110 | 12) 1101 | 16) 1111. |

Знаходимо коригувальні розряди для кожного інформаційного слова, як результат підсумовування за модулем 2 рядків перевірної матриці, номери яких збігаються з номерами одиниць в інформаційних розрядах коду. Наприклад, для інформаційного слова $I = [0111]$ (це п'ятнадцята комбінація) кодове слово має вигляд:

	101
Інформаційне слово $I=0111$	$\oplus 111$
	110
Результат підсумовування	<hr/> 100
Код, що передається	0111100

Аналогічно отримуємо кодові комбінації, що залишилися. Таким чином, будуємо повний код.

Процес декодування за синдромом здійснюється за допомогою перевірної матриці H . Для побудованого (7, 4) коду перевірна матриця має вигляд

$$H(7, 4) = \begin{vmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ a1 & a2 & a3 & a4 & p1 & p2 & p3 \end{vmatrix}$$

Для обраної комбінації маємо:

0	1	1	1	1	1	0
a1	a2	a3	a4	p1	p2	p3

Наприклад, відповідно до завдання $j_l=14$ червня ($14+6=20$), $n=7$. Тобто в процесі передачі відбулася помилка в 6-му інформаційному розряді.

$$20 \bmod 7 = 6$$

помилка 0111110 6 розряд

Відповідно до перевірної матриці складаємо перевірні вектори:

$$\begin{array}{llllll}
 p1 \oplus & a2 \oplus & a3 \oplus & a4 \oplus & =1 \oplus 1 \oplus 1 \oplus 1 & = 0 \\
 p2 \oplus & a1 \oplus & a3 \oplus & a4 \oplus & =1 \oplus 0 \oplus 1 \oplus 1 & = 1 \\
 p3 \oplus & a1 \oplus & a2 \oplus & a3 \oplus & =0 \oplus 0 \oplus 1 \oplus 1 & = 0
 \end{array}$$

Висновок: виявлено помилку. Синдром $S = 010$ (це шостий стовпчик перевірної матриці) показує, що помилка відбулася в 6-му інформаційному розряді, який необхідно інвертувати. Тобто, прийнятий код буде 0111100, що відповідає дійсності.

3.3.2. Зразок виконання завдання 2

Треба побудувати код Хеммінга для передачі кодової комбінації 11101101011. За заданою довжиною інформаційного слова ($k = 11$), використовуючи співвідношення (3.7), обчислимо основні параметри коду n та m .

$$m = \lceil \log_2 \{ (k+1) + \lceil \log_2 (k+1) \rceil \} \rceil = \lceil \log_2 \{ (11+1) + \lceil \log_2 (11+1) \rceil \} \rceil = 4,$$

при цьому $n = k + m = 14$, тобто отримали (14, 10)-код.

Визначаємо номери робочих і контрольних позицій кодової комбінації. Для розглянутого завдання (при $n = 14$) номери контрольних бітів будуть 1, 2, 4, 8. При цьому кодова комбінація має вигляд:

a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14
k	k	1	k	1	1	0	k	1	1	0	1	0	1

Запишемо перевірну матрицю H (14, 4):

$$H = \begin{array}{c|cccccccccccc|cc|c} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ \hline & a1 & a2 & a3 & a4 & a5 & a6 & a7 & a8 & a9 & a10 & a11 & a12 & a13 & a14 \end{array}$$

Проведемо перевірки на парність, починаючи з молодшого розряду. Визначимо значення $a1, a2, a4, a8$ контрольних бітів:

$$\begin{array}{llllllll} a1 \oplus & a3 \oplus & a5 \oplus & a7 \oplus & a9 \oplus & a11 \oplus & a13 & a1=1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 & =1 \\ a2 \oplus & a3 \oplus & a6 \oplus & a7 \oplus & a10 \oplus & a11 \oplus & a14 & a2=1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 & =0 \\ a4 \oplus & a5 \oplus & a6 \oplus & a7 \oplus & a12 \oplus & a13 \oplus & a14 & a4=1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 & =0 \\ a8 \oplus & a9 \oplus & a10 \oplus & a11 \oplus & a12 \oplus & a13 \oplus & a14 & a8=1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 & =0 \end{array}$$

Останній стовпчик містить значення перевірних бітів. Передана кодова комбінація буде така:

$$1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1$$

Припустимо, прийнято код:

$$1 \ 0 \ 1 \ 0 \ 1 \ \mathbf{0} \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1$$

Помилка є у 6-му розряді. Складемо синдром, щоб отримати номер помилкової позиції:

$$\begin{array}{llllllll} a1 \oplus & a3 \oplus & a5 \oplus & a7 \oplus & a9 \oplus & a11 \oplus & a13 \oplus & =1 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 & =\mathbf{0} \\ a2 \oplus & a3 \oplus & a6 \oplus & a7 \oplus & a10 \oplus & a11 \oplus & a14 \oplus & =0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 & =\mathbf{1} \\ a4 \oplus & a5 \oplus & a6 \oplus & a7 \oplus & a12 \oplus & a13 \oplus & a14 \oplus & =0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 & =\mathbf{1} \\ a8 \oplus & a9 \oplus & a10 \oplus & a11 \oplus & a12 \oplus & a13 \oplus & a14 \oplus & =0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 & =\mathbf{0} \end{array}$$

Синдром $S = 0110$ вказує на помилку у 6-му розряді.

Висновок: при декодуванні виявлена помилка, тому необхідно інвертувати 6-й розряд у кодовій комбінації. Тобто, отримаємо 10101100110101.

У лабораторній роботі досліджені засоби побудови групових кодів, що виправляють помилки в розрядах, які були при передачі повідомлень спотворені. Розглянуті процеси декодування з виправленням помилок.

ЛАБОРАТОРНА РОБОТА 4

ЛІНІЙНІ ЦИКЛІЧНІ КОДИ

Мета роботи: дослідити побудову та можливості корегування лінійних систематичних циклічних кодів.

4.1. Теоретичні відомості

Циклічні коди є основним засобом боротьби з помилками під час передачі інформації по каналах, де діють пакетні (групові) помилки розрядів. Вони дозволяють отримати високі якості коду в боротьбі з груповими помилками, мають відносно малі витрати на надлишковість і потребують прості, порівняно з іншими кодами, пристрої для кодування та декодування.

Циклічні коди є різновидом систематичних і мають ті ж загальні якості, крім своїх специфічних. Як і будь-який код, що виправляє помилки, циклічний має k інформаційних і m надлишкових (допоміжних чи перевірних розрядів), так що довжина комбінації становитиме $n = k + m$.

Для дослідження циклічних кодів використовують спеціальний математичний апарат, так звану *алгебру з остачею*.

Наведемо основні ідеї цього підходу в найпростішому вигляді.

Кожній кодовій n -розрядній комбінації $a_n a_{n-1} \dots a_2 a_1$ ставиться у відповідність алгебраїчний поліном $G(X)$, ступінь якого $n - 1$, причому

$$G(x) = a_n x^{n-1} + a_{n-1} x^{n-2} + \dots + a_2 x^1 + a_1 x^0, \quad (4.1)$$

де a_i – розрядні цифри коду, a_i для двійкового коду може бути нулем або одиницею;

x – фіктивна змінна.

Наприклад, кодовій комбінації 10110101 буде відповідати поліном $G(x) = x^7 + x^5 + x^4 + x^2 + 1$.

Зазначимо, що в поліномі $G(x)$ стільки членів, скільки одиниць у відповідному коді, тобто число членів дорівнює вазі коду.

Операції над поліномами та їх членами виконуються за правилами звичайної шкільної алгебри, за виключенням того, що замість плюса (+) і мінуса (-), діє плюс "за модулем два" (\oplus). Тобто там, де у звичайній алгебрі потрібно ставити мінус чи плюс, ставиться плюс "за модулем два" \oplus . Ця логічна функція повертає нуль, якщо підсумовується парне число одиниць (чи членів полінома з однаковим степенем x), проте якщо підсумовується непарне число одиниць – одиницю (або один x у відповідному степені).

У циклічних кодах дозволяються операції циклічного зміщення як вліво, так і вправо. Переміщення всіх розрядів коду на один крок (розряд) уперед еквівалентне множенню відповідного полінома $G(x)$ на x , а якщо розряди зміщувати на k позицій уперед, то це значить, що поліном $G(x)$ множиться на x^k .

Для побудови конкретного циклічного коду вибирають певним чином деякий поліном $P(x)$ степеню $m = n - k$. Цей поліном $P(x)$ назовемо *утворюючим*. Потім із усіх можливих поліномів, що мають степінь $n - 1$, добирають такі, що діляться без остачі на утворюючий поліном $P(x)$. Відібрані таким чином поліноми відповідають дозволеним комбінаціям n -розрядного циклічного коду.

Дійсно, циклічні коди мають поліноми, які без остачі діляться на утворюючий поліном. Якщо при діленні полінома прийнятого коду є остача, те це ознака того, що в кодовій комбінації сталися помилки і її можна забракувати, видавши відповідний сигнал про повторення передачі.

Від вигляду утворюючого полінома $P(x)$ суттєво залежать якості циклічного коду і, у першу чергу, його можливості щодо розпізнавання певної кількості помилок.

Проведемо деякі формальні операції, використавши поліном інформаційної частини $G(x)$ степеня $k - 1$ і утворюючий поліном $P(x)$ степеня m : помножимо $G(x)$ на X^m і поділимо добуток на утворюючий поліном $P(x)$. Тоді

$$\frac{X^m G(x)}{P(x)} = Q(x) \oplus \frac{R(x)}{P(x)}, \quad (4.1)$$

де $Q(x)$ – ціла частина від ділення $X^m G(x)$ на $P(x)$ того ж степеня, що і $G(x)$, а $R(x)$ – остача, причому має степінь, не вищий за $m - 1$.

Рівняння (4.1) перепишемо так:

$$P(x)Q(x) = x^m G(x) \oplus R(x). \quad (4.2)$$

Існують алгебраїчні та матричні способи побудови циклічних кодів. Рівняння (4.2) належить до алгебраїчного. Розрізняють два матричних способи, які мають канонічну та неканонічну форми представлення.

Канонічна форма запису матриці має вигляд як і у лінійному груповому коді. Вона складається з двох підматриць: одиничної транспонованої матриці I_k та перевірної додаткової R_{kn} . Визначаємо елементи додаткової матриці як остачі від ділення останнього рядка транспонованої матриці на утворюючий поліном. При цьому число остач повинне бути не менше k , число розрядів у остачі дорівнює m , а число одиниць у кожній остачі не менш $d - 1$. Якщо після приписування 0 до остачі отримуємо число коротше дільника, то записуємо дві остачі з нулем до і після залишку.

Перший спосіб: поліном інформаційного коду помножують на утворюючий. Наприклад, інформаційна частина – 101101 і їй відповідає поліном $G(x) = x^5 + x^3 + x^2 + 1$ (інформаційний поліном). Нехай утворюючий поліном буде $P(x) = x^3 + 1$. Тоді $P(x) \cdot G(x) = x^8 + x^6 + x^2 + 1$. Цьому поліному відповідає 9-розрядний циклічний код 101000101. Ясно, що це не роздільний код (невідомо де інформаційні, а де службові розряди). Цей недолік є причиною того, що такі коди не використовують на практиці.

Другий спосіб: інформаційні розряди зміщують на m розрядів вліво, а на m позицій, що залишилися, вписують код, що відповідає поліному остачі $R(x)$. У такому випадку чітко фіксовані позиції інформаційних та допоміжних розрядів, тобто код – роздільний, а тому з циклічних кодів тільки він і використовується для практичних цілей.

Наведемо приклад побудови коду, взявши інформаційну частину – 101001 і утворюючий поліном $P(x) = x^3 + 1$.

Тоді $x^m \cdot G(x) = x^3 \cdot (x^5 + x^3 + 1) = x^8 + x^6 + x^3$. Цьому поліному відповідає код 101001. Розділимо одержаний поліном на $P(x)$ і отримаємо

$R(x) = x^2$. Йому відповідає код остачі 100. Запишемо код остачі справа від отриманого вище коду 101001100.

Процес декодування проводять діленням отриманої кодової комбінації на утворюючий поліном. Якщо остача дорівнює нулю, передача повідомлення пройшла без спотворень. Якщо ні, проводять аналіз ваги остачі W (кількість одиниць у коді остачі, яка має бути менше або дорівнювати кількості помилок, що виправляють, $W \leq S$). Якщо умова не виконується, робиться циклічний зсув та повторюється ділення, цю процедуру повторюють доти, поки не виконається умова $W \leq S$. Останню остачу підсумовують з остачею і реалізують зворотний зсув на стільки розрядів, на скільки був попередній зсув.

4.2. Порядок виконання роботи

4.2.1. Вибір варіанта

j_1 = *прізвище, ім'я, по-батькові (сума кількості букв) – відповідає позиції помилкового символу та обчислюється як $j_1 \bmod (n)$, де n – кількість інформаційних символів у завданні;*

j_2 = *порядковий номер за списком – відповідає варіанту завдання.*

4.2.2 Послідовність дій

Лабораторна робота складається із двох завдань.

Відповідно до варіанта завдання виконати такі дії:

- показати процес кодування інформаційного слова;
- провести декодування слова;
- показати процес виправлення одиничної помилки для інформаційного слова;
- зробити висновки.

4.3. Зразок розрахунку завдань

4.3.1. Зразок виконання завдання 1 (варіант А)

Необхідно побудувати утворюючу матрицю циклічного коду, що виявляє всі одиничні помилки ($s=1$) при передачі k -розрядного інформаційного слова. Наведемо приклад розв'язання для $k = 7$.

Визначимо кількість перевірних розрядів.

Для $k = 7$ і $d = 2s + 1 = 3$ необхідне число перевірних розрядів

$$m = \lceil \log_2 \{ (k+1) + \lceil \log_2 (k+1) \rceil \} \rceil = \lceil \log_2 \{ (7+1) + \lceil \log_2 (7+1) \rceil \} \rceil = 4.$$

При цьому $n = k + m = 11$, тобто одержали (11, 7)-код.

За таблицею утворюючих багаточленів (дивись додаток 1) вибираємо поліном $P(x) = x^4 + x^3 + 1 = 11001$, тобто зі степенем, більшим або рівним m і числом ненульових членів, більшим або рівним d .

Будуємо транспоновану одиничну матрицю I_k .

$$I_k^T(11, 7) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Визначаємо елементи додаткової матриці як остачі від ділення останнього рядка транспонованої матриці на утворюючий поліном Далі будуємо утворюючу матрицю $G(n, k)$.

$$\begin{array}{r|l} \oplus \begin{array}{r} 1000000 \\ 11001 \\ \hline \end{array} & \begin{array}{l} 11001 \\ 111 \end{array} \\ \oplus \begin{array}{r} 10010 \\ 11001 \\ \hline \end{array} & \\ \oplus \begin{array}{r} 10110 \\ 11001 \\ \hline \end{array} & \\ \oplus \begin{array}{r} 11110 \\ 11001 \\ \hline \end{array} & 1010 \end{array} \quad R_{km}(11,7) = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

4.3.2. Зразок виконання завдання 1 (варіант Б)

Ще один спосіб побудови циклічного коду здійснюється шляхом множення одиничної матриці на утворюючий поліном. Вибирається утворюючий поліном $P(x)$ з таблиці багаточленів (див. додаток 1). Приклад роз'язання наводиться для коду (7, 4).

Для коду (7, 4) останній рядок одиничної матриці має вигляд 0 0 0 1.

Помноживши його на утворюючий поліном $P(x) = x^3 + x^2 + 1 = 1101$, отримаємо $0001 \cdot 1101 = 0001101$. Інші рядки отримаємо шляхом циклічного зсуву кодових комбінацій перетворювальної матриці.

$$G(7, 4) = \begin{vmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{vmatrix}$$

Рядки перетворювальної матриці являють собою 4 кодові комбінації, а інші 12 можуть бути отримані шляхом підсумовування за модулем два усіх можливих комбінацій рядків матриці.

4.3.2. Зразок завдання 2

Необхідно закодувати інформаційне слово циклічним кодом, що виправляє однократні помилки. Показати процес виправлення помилки.

Наведемо роз'язання для інформаційного полінома $G(x) = 1001$.

За заданим $k = 4$, для $s = 1$ визначимо довжину кодової комбінації n і кількість контрольних розрядів m коду за формулою

$$m = [\log_2 \{ (k+1) + [\log_2 (k+1)] \}] = [\log_2 \{ (4+1) + [\log_2 (4+1)] \}] = 3.$$

При цьому $n = k + m = 7$, тобто отримаємо $(7, 4)$ -код.

Будуємо інформаційний поліном до відповідного інформаційного слова довжиною k -біт

$$G(x) = x^3 + 1.$$

Здійснюємо зсув коду вліво на $m = n - k = 3$ розрядів, тобто поліном $G(x)$ множиться на x^m

$$G(x) \cdot x^m = (x^3 + 1) \cdot x^3 = x^6 + x^3 = 1001000.$$

Вибирається утворюючий поліном $P(x)$ за таблицею поліномів (див. додаток 1). Для виправлення одиничної помилки ($d = 3$) утворюючий поліном має бути ступеня $m = n - k = 3$, кількістю ненульових членів не менш $d = 3$ і входити у розкладання двочлена

$$x^7 + 1 = (x+1)(x^3 + x^2 + 1)(x^3 + x + 1).$$

Маємо два утворюючих поліноми третього степеня. Вибираємо один з них: $P(x) = x^3 + x + 1$.

Визначимо залишок $R(x)$ від розподілу $G(x) \cdot x^m$ на утворюючий поліном $P(x)$

$$\begin{array}{r} \oplus \begin{array}{r} x^6 + x^3 \\ x^6 + x^4 + x^3 \\ \hline x^4 \end{array} \bigg| \begin{array}{r} x^3 + x + 1 \\ x^3 + x \\ \hline 1 \end{array} \quad \oplus \begin{array}{r} 1001000 \\ 1011 \\ \hline 1010 \end{array} \\ \oplus \begin{array}{r} x^4 + x^2 + x \\ x^4 + x^2 + x \\ \hline x^2 + x \end{array} \quad \oplus \begin{array}{r} 1000 \\ 1011 \\ \hline 110 \end{array} \end{array}$$

Залишок $R(x) = x^2 + x = 110$.

Будуємо кодовий поліном $F(x)$, що передається.

$$F(x) = G(x) \cdot x^m + R(x) = x^6 + x^3 + x^2 + x = 1001110.$$

Розглянемо процедуру декодування, виявлення й виправлення помилки в прийнятій кодовій комбінації.

Припустимо, помилка відбулася в четвертому розряді кодової комбінації, при цьому повідомлення, що буде прийнято, має вигляд:

$$F^1(x) = F(x) + E(x) = 100 \boxed{0} 110.$$

Розділимо багаточлен $F^1(x)$, що відповідає отриманій кодовій комбінації, на утворюючий поліном, при цьому вага залишку (кількість одиниць у коді залишку) має бути меншою або дорівнювати кількості помилок, що виправляють, тобто $W \leq S$.

$$\begin{array}{r} F^1(x) = \oplus \begin{array}{r} 1000110 \\ 1011 \\ \hline 1111 \end{array} \bigg| \begin{array}{r} 1011 \\ 1011 \\ \hline 0 \end{array} \\ \oplus \begin{array}{r} 1111 \\ 1011 \\ \hline 1000 \end{array} \\ \oplus \begin{array}{r} 1000 \\ 1011 \\ \hline 11 \end{array} \quad W > S. \end{array}$$

Виконуємо циклічний зсув та повторюємо ділення, цю процедуру повторюємо доти, поки не виконається умова $W \leq S$.

$$1) \begin{array}{r|l} \oplus 0001101 & 1011 \\ \oplus 1011 & 1 \\ \hline & 110 \end{array}$$

$W > S.$

$$2) \begin{array}{r|l} \oplus 0011010 & 1011 \\ \oplus 1011 & 11 \\ \hline \oplus 1100 & \\ \oplus 1011 & \\ \hline & 111 \end{array}$$

$W > S.$

$$3) \begin{array}{r|l} \oplus 0110100 & 1011 \\ \oplus 1011 & 111 \\ \hline \oplus 1100 & \\ \oplus 1011 & \\ \hline \oplus 1110 & \\ \oplus 1011 & \\ \hline & 011 \end{array}$$

$W > S.$

$$4) \begin{array}{r|l} \oplus 1101000 & 1011 \\ \oplus 1011 & 1111 \\ \hline \oplus 1100 & \\ \oplus 1011 & \\ \hline \oplus 1110 & \\ \oplus 1011 & \\ \hline \oplus 1010 & \\ \oplus 1011 & \\ \hline & 1 \end{array}$$

$W = S.$

Підсумуємо остачу з останнім діленням:

$$\begin{array}{r} \oplus 1101000 \\ \oplus 001 \\ \hline 1101001 \end{array}$$

Здійснюємо зворотний циклічний зсув на 4 розряди отриманої комбінації 1101001

$$1110100 \rightarrow 0111010 \rightarrow 0011101 \rightarrow \boxed{1001}110.$$

Відкинувши контрольні розряди, отримуємо передане інформаційне слово.

Висновки: було досліджено засоби побудови циклічних кодів, що виправляють помилки, які виникають при передачі. Розглянуто процеси кодування і декодування, показано, як виправляється помилка.

ЛАБОРАТОРНА РОБОТА 5

СПОСОБИ ШИФРУВАННЯ ІНФОРМАЦІЇ

Мета роботи: дослідити способи побудови різних типів шифрів.

5.1. Теоретичні відомості

Існує два основних типи шифрування: з секретним ключем і з відкритим ключем. При шифруванні з секретним ключем потрібно, щоб усі сторони, які мають право на інформацію, мали однаковий ключ. Це дозволяє звести загальну проблему безпеки інформації до проблеми забезпечення захисту ключа. Шифрування з відкритим ключем є найбільш широко використовуваним методом шифрування. Він забезпечує конфіденційність інформації та гарантію того, що інформація залишається незмінною в процесі передачі, при цьому проблема збереження захисту ключа суттєво полегшується.

Повідомлення явного вигляду, що підлягає шифруванню, називають *відкритим текстом*, а результат шифрування – *шифр-текстом* або *криптограмою*. Керування процесом шифрування здійснюється деякою спеціальною інформацією – *ключем*. Захист інформації полягає в перетворенні її складових частин (слів, букв, цифр) у повідомлення неявного вигляду, для чого застосовуються керовані ключем спеціальні алгоритми, реалізовані як у програмному, так і в апаратному вигляді.

Шифрування з секретним ключем забезпечує конфіденційність інформації в зашифрованому стані. Будь-яка зміна в повідомленні, внесена під час передачі, буде виявлена, тому що після цього не вдасться правильно розшифрувати повідомлення. Шифрування з секретним ключем не забезпечує аутентифікацію, оскільки будь-який користувач може створювати, шифрувати і відправляти дійсне повідомлення, маючи доступ до секретного ключа.

Вся різноманітність перетворень основана на двох методах – *перестанови* і *підстанови*. Підстановки ще називають *замінами*. Дуже примітивним прикладом є *шифр Цезаря*. Літери відкритого тексту замінюються третьою (циклічною) буквою латинського алфавіту.

Наприклад,

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z	ключ з 25 літер
0 1 2 3 4 5 6 7 8 9	номер літери у відкритому тексті
C A E S A R	відкритий текст
2 0 4 18 0 17	секретний ключ (+ 3)
5 3 7 21 3 20	
F D H V D U	криптограма

Поліалфавітний підстановочний шифр із безліччю однобуквених ключів є шифром з періодичною заміною. До складу ключа може входити будь-який текст із загальнодоступної літератури (книги). Ключ підписується з повтореннями під повідомленнями (із циклічними повтореннями). Період шифру – кількість букв у циклі. Цифровий еквівалент кожної букви криптограми визначається у результаті додавання із приведенням за модулем m (кількість літер алфавіту) з еквівалентною літерою ключа.

Наприклад, в англійському алфавіті $m = 26$, тоді маємо:

C A R E F U L L Y	відкритий текст
2 0 17 4 5 20 11 11 24	
P I E S P I E S P	ключ
15 8 4 18 15 8 4 18 15	
R I V W V C P D N	криптограма
17 8 21 22 20 2 15 3 13	

Схема шифрування з більшою криптостійкістю основана на *таблиці Віжинера*. Шифр Віжинера із ключем, що складається з однієї букви, є *шифром Цезаря*, а з необмеженим ключем, що не повторюється, відомий як *шифр Вернама*. Цей шифр є найпоширенішим. Стійкість шифру підвищується за рахунок порушення статистичних зв'язків букв алфавіту. Шифр має високу надійність закриття тільки при використанні досить довгих ключів.

При перестановках змінюється не відкритий текст, а порядок символів. Вхідний потік ділиться на блоки, у кожному з яких виконується *перестановка символів*. Перестановки зазвичай отримують у результаті запису вихідного тексту й читання шифрованого тексту різними шляхами деякої геометричної фігури. Сукупність фігур, що забезпечують якісне маскування змісту блоку, зветься *маршрутами Гамільтона*, де послідовність заповнення вершин задається ключем.

Приклад перестановки символів:

1 2 3 4 5 6 7 8 9 10 11

E L E C T R O N I C S

відкритий текст

11 4 7 9 2 3 5 1 6 8 10

ключ

S C O I L E T E R N C

криптограма

Одноразові блокноти (One-time Pad, OTP) є єдиною теоретично системою шифрування, що не можна зламати. Одноразовий блокнот є списком чисел у випадковому порядку, який використовується для кодування повідомлення, виходячи з назви, один раз.

Шифрування з відкритим ключем є пізнішою технологією, ніж шифрування з секретним ключем. Головною відмінністю між цими двома технологіями є число ключів, використовуваних при шифруванні даних. У шифруванні з секретним ключем для шифрування і дешифрування даних використовується той самий ключ, тоді як в алгоритмах шифрування з відкритим використовуються два ключі. Один ключ використовується при шифруванні інформації, інший – при дешифруванні.

На практиці один з цих ключів називають *секретним*, а інший – *відкритим*. Секретний ключ тримається в таємниці власником пари цих ключів. Відкритий ключ передається разом з інформацією у відкритому вигляді. Принцип відкритого ключа: той, хто зашифрував текст, не обов'язково повинен бути здатний його розшифрувати. Недоліком систем шифрування з відкритим ключем є те, що вони вимагають великих обчислювальних потужностей а отже, є набагато менш швидкі.

Р. Ривест, А. Шамір, Л. Аделман запропонували різновид односторонніх функцій - функцію з потайним ходом RSA (за прізвищами авторів Riverst, Shamir, Adelman). Вони скористалися тим фактом, що знаходження простих чисел в обчислювальному відношенні здійснюється легко, але розкладання на множники двох таких чисел нездійсненно. Знаходження простих чисел основане на теоремі Ферма.

Розкриття такого шифру еквівалентно такому розкладанню. Тому для будь-якої довжини ключа можна дати оцінку нижньої межі кількості операцій для розкриття шифру, тобто гарантовано оцінити захищеність алгоритму.

Одностороння функція RSA – дискретне піднесення до степеня: $f_z(x) = x^e \pmod{n}$, де x – позитивне ціле, що не перевершує $n = p \cdot q$. Тут p і q – великі числа, такі, що функція Ейлера $\varphi(n) = (p-1)(q-1)$ має великий простий множник; $z = f(p, q, e)$ – потайний хід; e – позитивне ціле, яке не перевершує $\varphi(n)$. Причому найбільший спільний дільник (НСД) пари $(e, \varphi(n))$ дорівнює 1 (такі числа називають взаємно простими). Всім користувачам відомі n і e .

Зворотна функція $f_z^{-1}(y) = y^d \pmod{n}$, де d – єдине позитивне ціле, яке менше n і задовольняє умову $d \cdot e = 1 \pmod{\varphi(n)}$. Те, що $f_z^{-1}(y)$ – дійсно обернена функція y^d до x^e за модулем n , випливає з теореми Ейлера; а показник d визначають з алгоритму Евкліда, що обчислює НСД.

Зробити безпечним це можна за обчислення d за наявності відомих e і n . Мають на увазі, що власник пари ключів зберігає секретний ключ в таємниці і що відкритий ключ передається у відкритому вигляді. Отже, якщо інформація зашифрована за допомогою відкритого ключа, то дешифрувати її може тільки власник ключової пари.

При генеруванні ключів RSA необхідно дотримуватися ретельності. Щоб згенерувати ключову пару RSA, виконують такі кроки:

1. Обирають два простих числа p та q і тримають їх в секреті.
2. Обчислюють $n = p \cdot q$.
3. Розраховують $\varphi(n) = (p-1)(q-1)$.
4. Обирають таке e , щоб воно було *взаємно простим* по відношенню до $\varphi(n)$.
5. Визначають таке d , щоб $d \cdot e = 1 \pmod{\varphi(n)}$ і $d < \varphi(n)$.

Число n повинне містити близько 200 знаків або більше. Тоді обидва числа p і q повинні мати довжину, принаймні, 100 знаків. Ключі для використання на практиці повинні мати довжину 1024 біт. У випадку з секретною інформацією рекомендується використовувати ключі довжиною 2048 біт і більше.

Наведемо реальний приклад роботи алгоритму RSA. У даному випадку були вибрані числа $p = 11$ і $q = 13$. Тепер обчислюємо $n = p \cdot q$. Маємо $n = 11 \cdot 13 = 143$. Тепер потрібно обчислити $\varphi(n) = 10 \cdot 12 = 120$. Вибираємо чис-

ло e так, щоб воно було простим щодо $\varphi(n)$. Це означає, що НСД $(e, \varphi(n)) = 1$. Тут було вибрано $e = 7$. Визначаємо d так, щоб $d \cdot e = 1 \pmod{\varphi(n)}$ і $d < \varphi(n)$. Отже, $(d) \cdot 7 = 1 \pmod{120}$, а d повинне також бути менше за 120.

Знаходимо, що $d = 103$. Для цього 103 множимо на 7 і виходить 721. Ділимо 721 на 120 і отримуємо 6 із залишком 1. Отже, секретний ключ – (103, 143), а відкритим ключем буде (7, 143).

Для виконання безпосередньо шифрування і дешифрування використовуємо такі дії.

Шифрувальний алгоритм = (відкритий текст) $e \bmod n$.

Відкритий текст = (криптограма) $d \bmod n$.

Припустимо, що відкритий текст має вигляд $x = 9$. За допомогою формули шифрування отримуємо таке: $y = x^e \bmod n = 9^7 \bmod 143 = 48$.

При отриманні зашифрованої інформації вона піддається обробці алгоритмом дешифрування: $x = y^d \bmod n = 48^{103} \bmod 143 = 9$.

Майте на увазі, що тут використовуються числа, які відносно легко перевірити при виконанні даного прикладу. Насправді, в алгоритмі RSA використовуються набагато більші числа.

5.2. Порядок виконання роботи

5.2.1. Вибір варіанта

J - відкритий текст. Це може бути П.І.Б., число, місяць і рік народження або будь-яке повідомлення.

5.2.2. Послідовність дій

- Провести процес шифрування й дешифрування повідомлення з секретним і відкритим ключем.
- Зробити висновки.

УВАГА! В роботі використовують ПЕОМ (програма *CRIPTEP v4X*. або власна розробка криптоалгоритму, що вище оцінюється).

5.3. Зразок виконання завдання

Для шифрування необхідно розбити текст на блоки, кожен з яких може бути поданий числом $M(i) = 0, 1, \dots, n - 1$ нумерація букв в українському і латинському алфавітах починається з 1. Конвертація числа M в число E проводиться за правилом:

$$E(i) = (M(i) ^ e) \bmod n.$$

Для дешифрування використовуємо формулу:

$$M(i) = (E(i) ^ d) \bmod n.$$

Відповідно до алгоритму RSA підберемо параметри p, q, d, e .

Нехай $p = 17, q = 29$, тоді $n = 17 \cdot 29 = 493$. Вибираємо d взаємно просте з $\varphi(n) = (p-1)(q-1) = 448$, нехай $d = 13$. Підбираємо e так, що $(e \cdot d) \bmod \varphi(n) = 1$. $e = (1 + k \cdot \varphi(n)) / d$, де k – деяке ціле число. Так, для

$$k = 1: e = (1 + 1 \cdot 448) / 13 = 34,5384...$$

$$k = 2: e = (1 + 2 \cdot 448) / 13 = 69.$$

Відкритий ключ: (69, 493). Закритий ключ: (13, 493).

Шифрувальний алгоритм $E = M ^ e \bmod n$.

Нехай $M = \text{KASYANIK KATERINA}$.

Виконуючи шифрацію $E(i)$ (наприклад, для літери S обчислюємо $19^{69} \bmod 493 = 32$), послідовно заповнюємо таблицю.

Літера M_i	К	А	С	У	А	Н	І	К
№ з. п	11	1	19	25	1	14	9	11
Код E_i	282	1	32	94	1	437	42	282
Літера M_i	К	А	Т	Е	Р	І	Н	А
№ з. п	11	1	20	5	18	9	14	1
Код E_i	282	1	277	354	443	42	437	1

Тобто зашифроване повідомлення буде

$$E = 282\ 1\ 32\ 94\ 1\ 437\ 42\ 282\ 354\ 282\ 1\ 277\ 354\ 443\ 42\ 437\ 1.$$

Покажемо процес дешифрування. Наприклад, зашифроване повідомлення має вигляд:

$$E = 42\ 437\ 32\ 446\ 443\ 1\ 437\ 396\ 354.$$

Щоб отримати відкритий текст, треба розрахувати (криптограма) $d \bmod n$. Так, для першого коду маємо $42^{13} \bmod 493 = 9$. Це літера І. Повторюємо розрахунки для наступних кодів. Повне повідомлення має вигляд

$$M = \text{INSURANCE}.$$

Висновки: було проведено дослідження різних засобів шифрування і дешифрування.

6. РОЗРАХУНКОВА РОБОТА

ІНФОРМАЦІЙНІ ХАРАКТЕРИСТИКИ ДЖЕРЕЛА, ПРИЙМАЧА ТА КАНАЛУ ЗВ'ЯЗКУ З ПЕРЕШКОДАМИ

Мета роботи: навчитися розраховувати інформаційні характеристики системи передачі даних.

6.1. Теоретичні відомості

Під *системою зв'язку* будемо розуміти сукупність технічних засобів, що забезпечують передачу інформації з заданими властивостями від різних джерел різним одержувачам. Цілеспрямована розробка системи зв'язку може здійснюватися за умови наявності критеріїв ефективності її функціонування. Основним завданням системи зв'язку є забезпечення максимальної швидкості передачі при високій якості функціонування та економічності системи. Під якістю функціонування при цьому розуміють мінімізацію втрат інформації, що в підсумку трансформується в забезпечення високої правильності передачі.

Відомості, що є об'єктом зберігання, передачі і перетворення, називаються *інформацією*. Міра вимірювання кількості інформації основана на понятті ентропії. *Ентропія* – це міра невизначеності стану системи A (випадкової величини) з кінцевим або рахунковим числом результатів.

Ентропія, за Шенноном, $H(A)$ дорівнює сумі добутків ймовірностей станів системи і логарифмів цих ймовірностей, узятій зі зворотним знаком:

$$H(A) = -\sum_{i=1}^n p(a_i) \log_2 p(a_i), \quad (6.1)$$

де $A = (a_1, a_2, \dots, a_i, \dots, a_n)$ – безліч можливих станів системи, прийнятих елементами з ймовірностями $p(a_1), p(a_2), \dots, p(a_i), \dots, p(a_n)$;

n – число можливих станів.

У формулі (6.1) основа логарифма може бути двійковою, десятковою або натуральною. Якщо використовується двійкова основа, то вона може бути опущеною при запису. При двійковій основі ентропія вимірюється у

двійкових одиницях або бітах. Ентропія характеризує середнє значення кількості інформації і являє собою математичне очікування $-\log_2 p$, тобто

$$H(A) = M[-\log_2 p(a)] \quad (6.2)$$

Основними властивостями ентропії є:

1. Ентропія - це величина реальна, обмежена і не негативна: $H \geq 0$.
2. Ентропія мінімальна і дорівнює нулю, якщо хоча б один зі станів системи достеменно відомий: $H = H_{\min} = 0$.
3. Ентропія максимальна і рівна логарифму числа станів n , якщо стани системи рівноймовірні: $H = H_{\max} = \log_2 n$.
4. Ентропія бінарних величин змінюється від 0 до 1.

Ентропія дорівнює нулю, коли ймовірність одного із станів дорівнює нулю, потім зростає і досягає максимуму при $p = 0,36$ (для $n = 10$). При цьому невизначеність повідомлень при прийомі найбільша.

Визначимо *ентропію складної системи* $H(a,b)$, що представляє собою з'єднання декількох простих систем. Наприклад, є дві системи $A=(a_1, \dots, a_i, \dots, a_n)$ та $B=(b_1, \dots, b_j, \dots, b_m)$. При цьому можуть бути дві ситуації. Системи A і B можуть бути незалежними або залежними.

Розглянемо ситуацію для випадку незалежних систем. Для цього необхідно визначити ймовірності спільних подій $p_{ij} = p(a_i, b_j)$. Матриця ймовірностей має вигляд:

$$\begin{array}{ccccccc} p(b_1, a_1) & p(b_2, a_1) & \dots & p(b_m, a_1) \\ p(b_1, a_2) & p(b_2, a_2) & \dots & p(b_m, a_2) \\ \dots & \dots & \dots & \dots \\ p(b_1, a_m) & p(b_2, a_m) & \dots & p(b_m, a_m). \end{array} \quad (6.3)$$

При цьому дана матриця має властивості:

$$\sum_j p(a_i, b_j) = p(a_i), \quad \sum_i p(a_i, b_j) = p(b_j), \quad \sum_i p(a_i) = \sum_j p(b_j) = 1. \quad (6.4)$$

Після того, як відомі всі ймовірності, неважко обчислити ентропію.

$$H(A, B) = - \sum_{i=1}^n \sum_{j=1}^m p(a_i, b_j) \log_2 p(a_i, b_j). \quad (6.5)$$

Нехай системи A і B незалежні, тоді за теоремою множення ймовірностей (для незалежних випадкових величин) маємо, що при об'єднанні незалежних систем їх ентропії складаються:

$$p(a,b) = p(a) \cdot p(b) \text{ , } \log_2 p(a,b) = \log_2 p(a) + \log_2 p(b) \text{ ,} \quad (6.6)$$

$$H(A, B) = H(A) + H(B). \quad (6.7)$$

Нехай є дві залежні системи A і B . Позначимо умовну ймовірність $p\left(\frac{b_j}{a_i}\right)$ того, що система B набуде стан b_j за умови, що система A набула стану a_i . Визначимо умовну часткову ентропію системи B щодо окремої події a_i . При цьому повинні бути відомі умовні ймовірності $p\left(\frac{b_j}{a_i}\right)$.

Матриця ймовірностей має вигляд:

$$\begin{array}{ccccccc} p(b_1/a_1) & p(b_2/a_1) & \dots & p(b_m/a_1) \\ p(b_1/a_2) & p(b_2/a_2) & \dots & p(b_m/a_2) \\ \dots & \dots & \dots & \dots \\ p(b_1/a_m) & p(b_2/a_m) & \dots & p(b_m/a_m). \end{array} \quad (6.8)$$

Тоді часткова умовна ентропія буде дорівнювати

$$H(B/a_i) = \sum_{j=1}^m p(b_j/a_i) \log_2 p(b_j/a_i). \quad (6.9)$$

Щоб повністю охарактеризувати ентропію системи, необхідно визначити повну ентропію. Якщо часткову умовну ентропію усереднити по всіма станами a_i з урахуванням імовірності появи кожного зі станів $p(a_i)$, то знайдемо *повну умовну ентропію* повідомлень B відносно A .

$$H(B/A) = -\sum_{i=1}^n \sum_{j=1}^m p(a_i) \cdot p(b_j/a_i) \cdot \log_2 p(b_j/a_i). \quad (6.10)$$

Поняття умовної ентропії широко використовується для визначення інформаційних втрат при передачі інформації.

Нехай по каналу зв'язку передаються повідомлення за допомогою алфавіту A . У результаті впливу перешкод приймачем буде сприйматися інший алфавіт B . Ентропія $H(b_j/a_i)$ виражає невизначеність того що, відправивши a_i , отримаємо b_j , а поняття $H(a_j/b_i)$ – непевність, яка залишається після одержання b_j , у тому, що було відправлено саме a_i . Якщо в каналі зв'язку перешкоди відсутні, то завжди посланому символу a_1 відповідає прийнятий символ b_1 , $a_2 - b_2, \dots, a_n - b_n$. При цьому ентропія джерела $H(A)$ дорівнює ентропії приймача $H(B)$. Якщо в каналі зв'язку присутні перешкоди, то вони знищують частину переданої інформації.

Для обчислення ентропії спільної появи статистично залежних повідомлень використовують поняття ентропії об'єднання. Ентропія об'єднання обчислюється за формулою (6.5).

Ентропія об'єднання і умовна ентропія зв'язані між собою такими співвідношеннями:

$$H(A, B) = H(A) + H(A/B) = H(B) + H(B/A). \quad (6.11)$$

У випадку, якщо A і B між собою незалежні, то умовна ентропія дорівнює безумовній $H(B/A) = H(B)$, і тоді $H(A, B) = H(A) + H(B)$.

У загальному випадку ентропія об'єднаної системи становитиме $H(A, B) \leq H(A) + H(B)$, оскільки умовна ентропія менше безумовної $H(A/B) \leq H(B)$. Ентропія об'єднаної системи досягає максимуму тільки у випадку, якщо системи незалежні.

У випадку повної залежності систем стан однієї системи є станом іншої (вони еквівалентні): $H(A/B) = H(A) = H(B)$, оскільки $H(A/B) = 0$.

При передачі повідомлень відбувається зменшення невизначеності. Якщо про систему відомо все, то нема рації посилати повідомлення. І навпаки, чим стан системи був більше невизначеним, тим більшу кількість інформації отримуємо. Тому кількість інформації вимірюють зменшенням ентропії.

Припустимо, що отримана повна інформація про систему A . Нехай до спостереження ентропія системи дорівнювала H . У результаті спостереження ентропія дорівнює нулю $H = 0$, тобто про систему відомо все. Тоді інформація буде дорівнювати зменшенню ентропії: $I = H - 0$, тобто $I = H$.

На практиці часто буває, що система A для спостереження не доступна, і тоді ведуть спостереження за іншою системою B , пов'язаною із системою A . Між системою A і B є відмінності, які можуть бути двох видів:

1. Відмінності за рахунок того, що деякі стани системи A не знаходять відбиття в системі B (B менш повна, ніж система A).
2. Відмінності за рахунок помилок: неточностей виміру параметрів системи A і помилок при передачі повідомлень.

Виникає питання: яку кількість інформації про систему A дає спостереження системи B ? Дану інформацію визначають як зменшення ентропії системи A у результаті отримання відомостей про систему B :

$$I_{B \rightarrow A} = H(A) - H(A/B), \quad (6.12)$$

де $H(A)$ – апіорна ентропія до спостереження,

$H(A/B)$ – залишкова ентропія після отримання відомостей,

$I_{B \rightarrow A}$ – повна або середня інформація про систему A , що міститься в системі B .

Справедливою для системи B буде така формула:

$$I_{A \rightarrow B} = H(B) - H(B/A). \quad (6.13)$$

У загальному випадку, за наявності двох систем, кожна містить щодо іншої системи ту ж саму повну інформацію:

$I_{B \rightarrow A} = I_{A \rightarrow B} = I_{A \leftrightarrow B}$, а $I_{A \leftrightarrow B}$ називається *повною взаємною інформацією*, що втримується в системах A і B .

Це співвідношення для $I_{A \rightarrow B}$ наочно ілюструється на рис. 6.1.

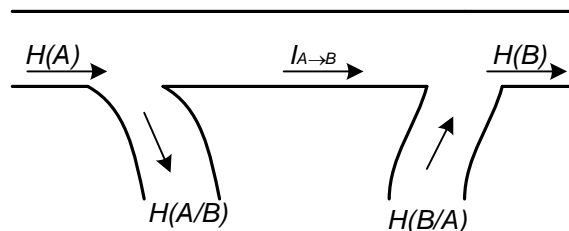


Рисунок 6.1 – Повна взаємна інформація

Тут ентропія джерела переданого сигналу – $H(A)$, тобто повна власна інформація, а $H(B)$ – ентропія прийнятого сигналу. Величина $H(A/B)$ являє

собою втрату інформації або ненадійність каналу, а $H(B/A)$ – показує створення помилкової, сторонньої інформації в каналі, яке не має відношення до джерела A і обумовлена присутніми в каналі перешкодами. За визначенням Шеннона, ненадійність каналу є ентропією входу, коли вихід відомий, тобто її можна вважати мірою середньої невизначеності прийнятого сигналу. Величина $H(B/A)$ є ентропія виходу, коли вхід відомий і служить мірою середньої невизначеності переданого сигналу. Співвідношення між $H(A/B)$ і $H(B/A)$ залежить від властивостей каналу.

Так, наприклад, при передачі звукового сигналу по каналу з вузькою смугою пропускання, недостатньою для якісного відтворення сигналу і з низьким рівнем перешкод, втрачається частина корисної інформації. В цьому випадку $H(A/B) \gg H(B/A)$. Якщо ж сигнал відтворюється якісно, але при цьому мають місце перешкоди від сусіднього каналу зв'язку, то це означає, що, майже не втрачаючи корисної інформації, отримується багато зайвої інформації та $H(A/B) \ll H(B/A)$.

Нехай кількість інформації, яка передається по каналу зв'язку за час T , дорівнює

$$I_T = H_T(A) - H_T(A/B). \quad (6.14)$$

Якщо передача повідомлення триває T одиниць часу, то швидкість передачі інформації становитиме

$$R = \frac{I_T}{T} = \frac{1}{T} (H_T(A) - H_T(A/B)) = H(A) - H(A/B). \quad (6.15)$$

Це кількість інформації, що доводиться в середньому на одне повідомлення. Якщо в секунду передається k повідомлень, то швидкість передачі буде становити

$$R = k \cdot [H(A) - H(A/B)]. \quad (6.16)$$

Пропускна здатність каналу є максимально досяжною швидкістю передачі інформації для даного каналу:

$$C = \max R = k \cdot [H(A) - H(A/B)]_{\max}. \quad (6.17)$$

Інформаційна швидкість або швидкість передачі інформації визначається середньою кількістю інформації, яка передається за одиницю часу і вимірюється у біт/с. Для повідомлень, що складені з рівноймовірних взаємно незалежних символів,

$$R = \frac{1}{\tau} \log_2 m, \quad (6.18)$$

де τ – середній час вироблення одного символу.

У випадку, якщо символи не рівно ймовірні,

$$R = -\frac{1}{\tau} \sum_i p_i \log_2 p_i. \quad (6.19)$$

Для символів, що мають різну тривалість,

$$R = \frac{-\sum_i p_i \cdot \log_2 p_i}{\sum_i \tau_i \cdot p_i}. \quad (6.20)$$

Для розрахунку пропускну здатності треба записати у чисельник максимальну ентропію:

$$C_{\max} = \frac{H_{\max}}{\tau}. \quad (6.21)$$

Припустимо, що маємо дискретний канал, ймовірність виникнення помилки в якому близька до нуля. Такий канал називають ідеальним каналом або каналом без шуму. Пропускна здатність каналу визначається відповідно до (6.21). За наявності ідеального каналу природно поставити питання про можливість передачі по ньому без втрат інформації від довільного дискретного джерела A , що характеризується продуктивністю \bar{H} зі швидкістю, яка дорівнює пропускну здатності каналу. Схема побудови такої системи передачі інформації повинна виглядати так, як на рис. 6.2. Для того щоб швидкість передачі інформації в каналі дорівнювала його пропускну здатності, на вході каналу повинно бути присутнє дискретне джерело з певними статистичними властивостями, що максимізує величину $I_{A \rightarrow B}$.

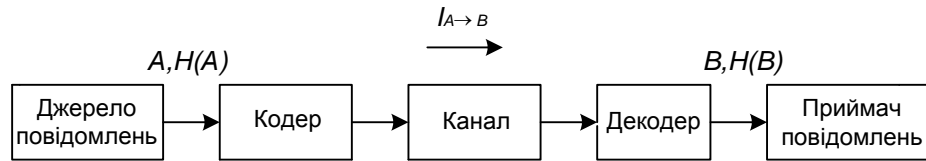


Рисунок 6.2 – Система передачі інформації

Зокрема, такому джерелу треба мати максимальну ентропію або нульову надмірність, тобто видавати незалежні рівноймовірні повідомлення. У той же час бажано мати можливість передавати повідомлення від довільного джерела з будь-якими статистичними властивостями, тобто такого, що має ненульову надмірність. Таким чином, функцією кодера є узгодження в статичному сенсі повідомлень джерела зі входом каналу.

Пропускна здатність є найважливішою характеристикою каналів зв'язку. Виникає питання: яка має бути пропускна здатність каналу, щоб інформація від джерела A до приймача B надходила без затримок? Відповідь на це питання дає перша теорема Шеннона.

Коли є джерело інформації з ентропією $H(A)$ і канал зв'язку із пропускну здатністю C , то якщо $C > H(A)$, завжди можна закодувати досить довге повідомлення таким чином, що воно буде передано без затримок. Якщо ж, навпаки, $C < H(A)$, то передача інформації без затримок неможлива.

Продуктивність джерела повідомлень є ентропією за одиницю часу:

$$\bar{H} = \frac{k \cdot H(A)}{\tau}. \quad (6.22)$$

У будь-якому реальному каналі завжди присутні перешкоди. Однак якщо їх рівень настільки малий, що імовірність помилковості практично дорівнює нулю, можна умовно вважати, що всі сигнали передаються неспотвореними. У цьому випадку середня кількість інформації дорівнює $I(A, B) = I(B, A) = H(A)$. Максимальне значення $H_{\max} = \log_2 m$.

Для дискретних каналів з перешкодами Шеннон дав другу теорему.

Нехай є джерело інформації X , ентропія якого в одиницю часу дорівнює $H(A)$, і канал із пропускну здатністю C . Якщо $H(A) > C$, то при будь-якому кодуванні передача повідомлень без затримок і спотворень не-

можлива. Якщо ж $H(A) < C$, то будь-яке досить довге повідомлення можна завжди закодувати так, що воно буде передано без затримок і спотворень із ймовірністю наскільки завгодно близькою до одиниці.

Однією з інформаційних характеристик джерела дискретних повідомлень є надмірність, яка визначає, яка частка максимально можливої ентропії не використовується джерелом повідомлень. Якщо ентропія джерела повідомлень не дорівнює максимальній ентропії для алфавіту з даною кількістю якісних ознак (мають на увазі якісні ознаки алфавіту, за допомогою яких складаються повідомлення), то це передусім означає, що повідомлення даного джерела могли б нести більшу кількість інформації. Абсолютне недовантаження на символ повідомлень визначається як

$$\Delta D = (H_{\text{макс}} - H) . \quad (6.23)$$

Інформаційна надмірність показує відносне недовантаження на символ алфавіту і є безрозмірною величиною:

$$D = 1 - \frac{H(A)}{H_{\text{max}}(A)} = 1 - \mu , \quad (6.24)$$

де μ – коефіцієнт стискання.

6.2. Порядок виконання роботи

6.2.1. Вибір варіанта

Для вибору свого варіанта необхідно задати такі дані:

j_1 = сума цифр – день народження + рік народження – відповідає кількості переданих символів;

j_2 = місяць народження – відповідає середньому часу виробітку символу на виході джерела в мілісекундах;

j_3 = порядковий номер за списком – відповідає номеру матриці спільних ймовірностей $p(A, B)$, взятого з переліку варіантів завдання.

6.2.2 Відповідно до свого варіанта завдання виконати такі розрахунки

- Розрахувати інформаційні характеристики дискретного джерела повідомлень: ентропію $H(A)$, продуктивність \bar{N} , надмірність D .
- Розрахувати інформаційні характеристики дискретного приймача повідомлень: ентропію $H(B)$.
- Розрахувати необхідні інформаційні характеристики каналу зв'язку. Побудувати каналні матриці з боку джерела та з боку приймача. Розрахувати: ентропію $H(A/B)$, $H(B/A)$, $H(A,B)$; кількість інформації I ; втрати інформації ΔI ; швидкість інформації R ; пропускну здатність каналу C .
- Зробити висновки за результатами роботи.

УВАГА! Всі розрахунки виконуються на комп'ютері.

6.3. Зразок розрахунку

1) Для заданої матриці спільних ймовірностей $p(A,B)$ обчислимо безумовні ймовірності як суми спільних ймовірностей по рядках та по стовпцях матриці:

$$p(A,B) = \begin{array}{c|ccc} & 0 & 0 & 0,02 \\ \hline 0,1 & 0,08 & 0,1 & \\ 0,1 & 0,1 & 0,5 & \end{array} \begin{array}{l} p(a_i) \\ 0,02 \\ 0,28 \\ 0,7 \end{array}$$
$$p(b_j) \quad 0,2 \quad 0,18 \quad 0,62$$

2) Розрахуємо ентропію джерела A і приймача B , використовуючи формулу

$$H(A) = -\sum p(a_i) \log_2 p(a_i),$$
$$H(A) = -(0,02 \log_2 0,02 + 0,28 \log_2 0,28 + 0,7 \log_2 0,7) = 0,987 \text{ біт/симв},$$
$$H(B) = -\sum p(b_j) \log_2 p(b_j),$$
$$H(B) = -(0,2 \log_2 0,2 + 0,18 \log_2 0,18 + 0,62 \log_2 0,62) = 1,337 \text{ біт/симв}.$$

3) Визначимо умовні ймовірності та побудуємо матрицю умовних ймовірностей з боку приймача:

$$p(a_i / b_j) = \frac{p(a_i, b_j)}{p(b_j)};$$

$$p(a_1 / b_1) = \frac{0}{0,2} = 0; \quad p(a_1 / b_2) = \frac{0}{0,18} = 0;$$

$$p(a_1 / b_3) = \frac{0,02}{0,62} = 0,032;$$

$$p(a_2 / b_1) = \frac{0,1}{0,2} = 0,5; \quad p(a_2 / b_2) = \frac{0,08}{0,18} = 0,444;$$

$$p(a_2 / b_3) = \frac{0,1}{0,62} = 0,161;$$

$$p(a_3 / b_1) = \frac{0,1}{0,2} = 0,5; \quad p(a_3 / b_2) = \frac{0,1}{0,18} = 0,556;$$

$$p(a_3 / b_3) = \frac{0,5}{0,62} = 0,806.$$

Заповнюємо матрицю умовних ймовірностей з боку приймача:

$$p(a_i / b_j) = \begin{vmatrix} 0 & 0 & 0,032 \\ 0,5 & 0,444 & 0,161 \\ 0,5 & 0,556 & 0,806 \end{vmatrix}.$$

Визначимо повну умовну ентропію $H(A/B)$, використовуючи формулу:

$$H(A/B) = -\sum_i \sum_j p(b_j) \cdot p(a_i / b_j) \cdot \log_2 p(a_i / b_j),$$

$$H(A/B) = -[0,2(2 \cdot 0,5 \log_2 0,5) + 0,18(0,444 \log_2 0,444 + 0,556 \log_2 0,556) + \\ + 0,62(0,032 \log_2 0,032 + 0,161 \log_2 0,161 + 0,806 \log_2 0,806)] = 0,896 \text{ біт/симв}$$

або враховуючи, що $p(a_i, b_j) = p(b_j) \cdot p(a_i / b_j)$, маємо:

$$H(A/B) = -\sum_i \sum_j p(a_i, b_j) \cdot \log_2 p(a_i / b_j),$$

$$H(A/B) = -(2 \cdot 0,1 \cdot \log_2 0,5 + 0,08 \cdot \log_2 0,444 + 0,1 \cdot \log_2 0,556 + 0,02 \cdot \log_2 0,032 + \\ + 0,1 \cdot \log_2 0,161 + 0,5 \cdot \log_2 0,806) = 0,896 \text{ біт/симв}.$$

Результати повністю збігаються.

4) Аналогічно виконаємо розрахунки щодо джерела повідомлень.

Побудуємо каналну матрицю $p(b_j / a_i)$. Обчислимо формулу

$$p(b_j / a_i) = \frac{p(a_i, b_j)}{p(a_i)} \quad \text{і заповнимо рядки матриці умовних ймовірностей з боку джерела повідомлень:}$$

$$p(b_j / a_i) = \begin{vmatrix} 0 & 0 & 1 \\ 0,357 & 0,286 & 0,357 \\ 0,142 & 0,142 & 0,714 \end{vmatrix}.$$

Розрахуємо умовну ентропію:

$$H(B / A) = - \sum_i \sum_j p(a_i) \cdot p(b_j / a_i) \cdot \log_2 p(b_j / a_i),$$

$$H(B / A) = -[0,28(0,357 \cdot \log_2 0,357 + 0,286 \cdot \log_2 0,286 + 0,357 \cdot \log_2 0,357) + 0,7(0,142 \cdot \log_2 0,142 + 0,142 \cdot \log_2 0,142 + 0,714 \cdot \log_2 0,714)] = 1,246 \text{ біт/симв}$$

$$\text{або } H(B / A) = - \sum_i \sum_j p(a_i, b_j) \cdot \log_2 p(b_j / a_i),$$

$$H(B / A) = -(0,1 \cdot \log_2 0,357 + 0,1 \cdot \log_2 0,142 + 0,08 \cdot \log_2 0,286 + 0,1 \cdot \log_2 0,142 + 0,02 \cdot \log_2 1 + 0,1 \cdot \log_2 0,357 + 0,5 \cdot \log_2 0,714) = 1,246 \text{ біт/симв.}$$

5) Визначимо ентропію об'єднання, використовуючи формулу

$$H(A, B) = - \sum_i \sum_j p(a_i, b_j) \cdot \log_2 p(a_i, b_j),$$

$$H(A, B) = -(4 \cdot 0,1 \cdot \log_2 0,1 + 0,08 \cdot \log_2 0,08 + 0,02 \cdot \log_2 0,02 + 0,5 \cdot \log_2 0,5) = 2,233 \text{ біт/симв.}$$

Зробимо перевірку:

$$H(A, B) = H(A) + H(B / A),$$

$$H(A, B) = 0,987 + 1,246 = 2,233 \text{ біт/симв,}$$

$$H(B, A) = H(B) + H(A / B),$$

$$H(A, B) = 1,337 + 0,896 = 2,233 \text{ біт/симв.}$$

6) Знайдемо втрати інформації в каналі зв'язку:

З боку джерела (відплив інформації)

$$\Delta I_A = k \cdot H(B/A),$$

$$\Delta I_A = 1994 \cdot 1,246 = 2485 \text{ біт.}$$

З боку приймача (стороння інформація)

$$\Delta I_B = k \cdot H(A/B),$$

$$\Delta I_B = 1994 \cdot 0,896 = 1787 \text{ біт.}$$

7) Визначимо кількість взаємної інформації на k переданих символів:

$$I(A, B) = k \cdot [H(B) - H(B/A)] = k \cdot H(B) - \Delta I_A,$$

$$I(A, B) = 1994 \cdot 1,337 - 2485 = 181 \text{ біт,}$$

$$I(B, A) = k \cdot [H(A) - H(A/B)] = k \cdot H(A) - \Delta I_B,$$

$$I(B, A) = 1994 \cdot 0,987 - 1787 = 181 \text{ біт.}$$

8) Зробимо перевірку. Знайдемо кількість інформації при передачі повідомлень по каналу зв'язку з перешкодами безпосередньо з матриці об'єднання:

$$I(B, A) = k \cdot [H(A) + H(B) - H(B, A)],$$

$$I(B, A) = 1994 \cdot (1,337 + 0,987 - 2,233) = 181 \text{ біт.}$$

9) Розрахуємо швидкість передачі інформації R (інформаційний потік).

$$R = \frac{I(A, B)}{\tau} = \frac{I(B, A)}{\tau}$$

$$R = \frac{181}{12 \cdot 10^{-3}} \approx 15 \text{ кбіт/с.}$$

10) Визначимо пропускну здатність каналу C

$$C = k \cdot \max \left\{ \frac{I(A, B)}{\tau} \right\} = k \frac{\log m}{\tau},$$

де $m=3$, кількість переданих (прийнятих) символів.

$$C = \frac{1994 \cdot 1,584}{0,012} \approx 263 \text{ кбіт/с.}$$

11) Знайдемо продуктивність джерела A

$$\overline{H} = \frac{k \cdot H(A)}{\tau},$$

$$\overline{H} = \frac{1994 \cdot 0,987}{0,012} \approx 164 \text{ кбіт/с.}$$

Таким чином, $\overline{H} \leq C$, тобто виконується умова кодування каналу.

12) Надмірність джерела D

$$D = 1 - \frac{H(A)}{H_{\max}(A)} = 1 - \mu.$$

$$D = 1 - \frac{0,987}{1,584} = 0,377 \approx 38 \text{ \%}.$$

13) Висновки: у роботі були зроблені розрахунки параметрів системи передачі інформації, а саме джерела та приймача повідомлень, каналу зв'язку, дана оцінка продуктивності та надмірності джерела повідомлень.

6.4 Порядок оформлення розрахункової роботи

Пояснювальна записка повинна містити:

- бланк завдання;
- титульний аркуш;
- зміст;
- вступ;
- основну частину;
- висновки;
- список джерел інформації;
- додатки (за необхідності).

Титульний аркуш розрахункової роботи містить:

- назву університету та кафедри, де було виконано розрахункове завдання;
- прізвище, ім'я, по батькові студента;
- найменування роботи;
- шифр і найменування спеціальності;
- науковий ступінь, вчене звання, прізвище та ініціали керівника;
- місто і рік.

Варіант розрахункової роботи уточнюється у викладача (за списками).

СПИСОК ЛІТЕРАТУРИ

1. Дмитриев В.И. Прикладная теория информации / В.И. Дмитриев. – К.: Вища шк., 1989. – 320 с.
2. Цимбал У.П. Теория информации и кодирования / У.П. Цимбал. – К.: Вища шк., 1992. – 264 с.
3. Вернер М. Основы кодирования : учебник для вузов. – М.: Техносфера, 2004. – 286 с.
4. Лидовский В.В. Теория информации : учеб. пособ. – М.: Компания Спутник+, 2004. – 111 с.
5. Кудряшов Б.Д. Теория информации : учебник для вузов. – СПб.: Питер, 2009. – 320 с.
6. Золотарев В. В. Помехоустойчивое кодирование. Методы и алгоритмы : справочник / В. В Золотарев., Г. В. Овечкин; под ред. чл.-кор. РАН Ю.Б. Зубарева. – Горячая линия – Телеком, 2004. – 126 с.
7. Осипян В.О. Криптография в задачах и упражнениях / В.О. Осипян, К.В. Осипян. – М.: Гелиос, АРВ, 2004. – 144 с.
8. Вентцель Е.С. Теория вероятностей и ее инженерные приложения / Е.С. Вентцель, Л.А. Овчаров. – 2-е изд., стер. – М.: Высш. шк., 2000. – 480 с.

ДОДАТОК 1

Фрагмент таблиці утворюючих багаточленів

Код	Поліном	Код	Поліном
11	$x+1$	1000001	x^6+1
101	x^2+1	1000011	x^6+x+1
111	x^2+x+1	1000101	x^6+x^2+1
1001	x^3+1	1000111	x^6+x^2+x+1
1011	x^3+x+1	1001001	x^6+x^3+1
1101	x^3+x^2+1	1001011	x^6+x^3+x+1
1111	x^3+x^2+x+1	1001101	$x^6+x^3+x^2+1$
10001	x^4+1	1001111	$x^6+x^3+x^2+x+1$
10011	x^4+x+1	1010001	x^6+x^4+1
10101	x^4+x^2+1	1010011	x^6+x^4+x+1
10111	x^4+x^2+x+1	1010101	$x^6+x^4+x^2+1$
11001	x^4+x^3+1	1010111	$x^6+x^4+x^2+x+1$
11011	x^4+x^3+x+1	1011001	$x^6+x^4+x^3+1$
111101	$x^5+x^4+x^3+x^2+1$	1011011	$x^6+x^4+x^3+x+1$
111111	$x^5+x^4+x^3+x^2+x+1$	1011101	$x^6+x^4+x^3+x^2+1$
100001	x^5+1	1011111	$x^6+x^4+x^3+x^2+x+1$
100011	x^5+x+1	1100001	x^6+x^5+1
100101	x^5+x^2+1	1100011	x^6+x^5+x+1
100111	x^5+x^2+x+1	1100101	$x^6+x^5+x^2+1$
101001	x^5+x^3+1	1100111	$x^6+x^5+x^2+x+1$
101011	x^5+x^3+x+1	1101001	$x^6+x^5+x^3+1$
101101	$x^5+x^3+x^2+1$	1101011	$x^6+x^5+x^3+x+1$
101111	$x^5+x^3+x^2+x+1$	1101101	$x^6+x^5+x^3+x^2+1$
110001	x^5+x^4+1	1101111	$x^6+x^5+x^3+x^2+x+1$
110011	x^5+x^4+x+1	1110001	$x^6+x^5+x^4+1$
110101	$x^5+x^4+x^2+1$	1110011	$x^6+x^5+x^4+x+1$
110111	$x^5+x^4+x^2+x+1$	1110101	$x^6+x^5+x^4+x^2+1$
111001	$x^5+x^4+x^3+1$	1110111	$x^6+x^5+x^4+x^2+x+1$
111011	$x^5+x^4+x^3+x+1$	1111001	$x^6+x^5+x^4+x^3+1$
111101	$x^5+x^4+x^3+x^2+1$	1111011	$x^6+x^5+x^4+x^3+x+1$
111111	$x^5+x^4+x^3+x^2+x+1$	1111101	$x^6+x^5+x^4+x^3+x^2+1$
		1111111	$x^6+x^5+x^4+x^3+x^2+x+1$

ЗМІСТ

Вступ	3
Лабораторна робота 1. Перетворення форми інформації. Аналого-цифрове та цифро-аналогове перетворення. Розрахунок кількості інформації	4
Лабораторна робота 2. Оптимальне кодування	18
Лабораторна робота 3. Перешкодостійке кодування. Лінійні групові коди	27
Лабораторна робота 4. Лінійні циклічні коди	37
Лабораторна робота 5. Способи шифрування інформації	45
6. Розрахункова робота. Інформаційні характеристики джерела, приймача та каналу зв'язку з перешкодами	51
Список літератури	65
Додаток 1. Фрагмент таблиці утворюючих багаточленів	66

Навчальне видання

ФЕТЮХІНА Людмила Вікторівна
БУТОВА Ольга Анатоліївна

ТЕОРІЯ ІНФОРМАЦІЇ ТА КОДУВАННЯ

Навчально - методичний посібник

Роботу рекомендував до видання проф. В. Т. Долбня

Редактор – О.С. Самініна

План 2012 р., поз. 111/

Формат 60х84 1/16. Папір офсетн. Друк – ризографія.

Гарнітура – Times New Roman. Наклад 50 прим. Ціна договірна.

Видавничий центр НТУ «ХП».

Свідоцтво про державну реєстрацію ДК № 3657 від 24.12.2009 р.
61002, м. Харків, вул. Фрунзе, 21

Друкарня НТУ «ХП».
61002, Харків, вул. Фрунзе, 21